
PLay: Efficient Topological Layer based on Persistence Landscapes

Kwangho Kim

Carnegie Mellon University
Pittsburgh, USA
kwanghk@cmu.edu

Jisu Kim

Inria
Palaiseau, France
jisu.kim@inria.fr

Manzil Zaheer

Google Research
Mountain View, USA
manzilzaheer@google.com

Joon Sik Kim

Carnegie Mellon University
Pittsburgh, USA
joonsikk@cs.cmu.edu

Frederic Chazal

Inria
Palaiseau, France
frederic.chazal@inria.fr

Larry Wasserman

Carnegie Mellon University
Pittsburgh, USA
larry@stat.cmu.edu

Abstract

We propose PLLay, a novel topological layer for general deep learning models based on persistence landscapes, in which we can efficiently exploit the underlying topological features of the input data structure. In this work, we show differentiability with respect to layer inputs, for a general persistent homology with arbitrary filtration. Thus, our proposed layer can be placed anywhere in the network and feed critical information on the topological features of input data into subsequent layers to improve the learnability of the networks toward a given task. A task-optimal structure of PLLay is learned during training via backpropagation, without requiring any input featurization or data preprocessing. We provide a novel adaptation for the DTM function-based filtration, and show that the proposed layer is robust against noise and outliers through a stability analysis. We demonstrate the effectiveness of our approach by classification experiments on various datasets.

1 Introduction

With its strong generalizability, deep learning has been pervasively applied in machine learning. To improve the learnability of deep learning models, various techniques have been proposed. Some of them have achieved an efficient data processing method through specialized layer structures; for instance, inserting a convolutional layer greatly improves visual object recognition and other tasks in computer vision [e.g., Krizhevsky et al., 2012, LeCun et al., 2016]. On the other hand, a large body of recent work focuses on optimal architecture of deep network [Simonyan and Zisserman, 2015, He et al., 2016, Szegedy et al., 2015, Albelwi and Mahmood, 2016].

In this paper, we explore an alternative way to enhance the learnability of deep learning models by developing a novel *topological layer* which feeds the significant topological features of the underlying data structure in an arbitrary network. The power of topology lies in its capacity which differentiates sets in topological spaces in a robust and meaningful geometric way [Carlsson, 2009, Ghrist, 2008]. It provides important insights into the global "shape" of the data structure via *persistent homology*

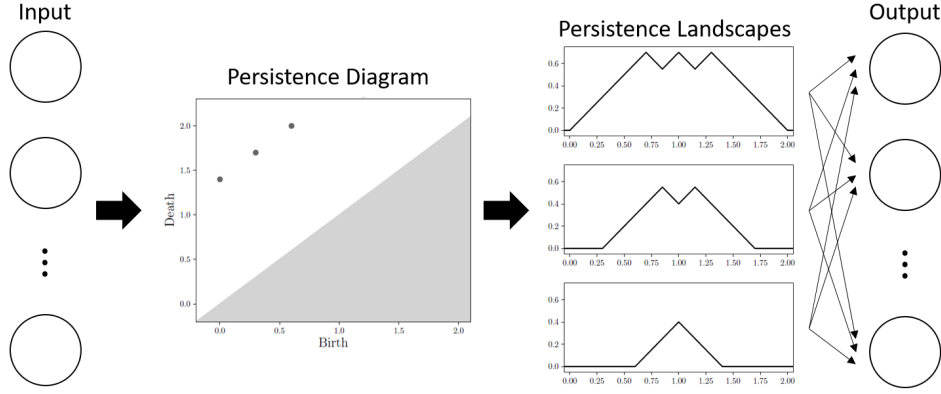


Figure 1: Illustration of PLLay, a novel topological layer based on weighted persistence landscapes. Information in the persistence diagram is first encoded into persistence landscapes as a form of vectorized function, and then a deep learning model determines which components of the landscape (e.g., particular hills or valleys) are important for a given task during training. PLLay can be placed anywhere in the network.

[Zomorodian and Carlsson, 2005]. The use of topological methods in data analysis has been limited by the difficulty of combining the main tool of the subject, persistent homology, with statistics and machine learning. Nonetheless, a series of recent studies have reported notable successes in utilizing topological methods in data analysis [e.g., Zhu, 2013, Dindin et al., 2020, Nanda and Sazdanović, 2014, Tralie and Perea, 2018, Seversky et al., 2016, Gamble and Heo, 2010, Pereira and de Mello, 2015, Umeda, 2017, Liu et al., 2016, Venkataraman et al., 2016, Emrani et al., 2014]

There are at least three benefits of utilizing the topological layer in deep learning; 1) we can efficiently extract robust global features of input data that otherwise would not be readily accessible via traditional feature maps, 2) an optimal structure of the layer for a given task can be easily embodied via backpropagation during training, and 3) with proper filtrations it can be applied to arbitrarily complicated data structure even without any data preprocessing.

Related Work. The idea of incorporating topological concepts into deep learning has been explored only recently, mostly via feature engineering perspective where we use some fixed, predefined features that contain topological information [e.g., Dindin et al., 2020, Umeda, 2017, Liu et al., 2016]. Guss and Salakhutdinov [2018], Rieck et al. [2019] proposed a complexity measure for neural network architectures based on topological data analysis. Carlsson and Gabrielsson [2020] applied topological approaches to deep convolutional networks to understand and improve the computations of the network. Hofer et al. [2017] first developed a technique to input persistence diagrams into neural networks by introducing their own topological layer. Carrière et al. [2020] proposed a network layer for persistence diagrams built on top of graphs. Poulenard et al. [2018], Gabrielsson et al. [2019], Hofer et al. [2019], Moor et al. [2020] also proposed various topology loss functions and layers applied to deep learning. Nevertheless, all the previous approaches suffer from at least one or more of the following limitations: 1) they rely on a particular parametrized map or filtration, 2) they lack stability results or the stability is limited to a particular type of input data representation, and 3) most importantly, the differentiability of persistent homology is not guaranteed with respect to the layer’s input therefore we can not place the layer in the middle of deep networks in general.

Contribution. This paper presents a new topological layer, PLLay (Persistence Landscape-based topological Layer: see Figure 1 for an illustration), that does not suffer from the above limitations. Our topological layer does not rely on a particular filtration or a parametrized mapping but still shows favorable theoretical properties. The proposed layer is designed based on the weighted persistence landscapes to be less prone to extreme topological distortions. We provide a tight stability bound that does not depend on the input complexity, and show the stability with respect to input perturbations. We also provide a novel adaptation for the DTM function-based filtration, and analyze the stability property. Importantly, we guarantee the differentiability of our layer with respect to the layer’s input.

Reproducibility. The code for PLLay is available at <https://github.com/jisuk1/pllay/>.

2 Background and definitions

Topological data analysis (TDA) is a recent and emerging field of data science that relies on topological tools to infer relevant features for possibly complex data [Carlsson, 2009]. In this section, we briefly review basic concepts and main tools in TDA which we will harness to develop our topological layer in this paper. We refer interested readers to Chazal and Michel [2017], Hatcher [2002], Edelsbrunner and Harer [2010], Chazal et al. [2009, 2016b] for details and formal definitions.

2.1 Simplicial complex, persistent homology, and diagrams

When inferring topological properties of \mathbb{X} , a subset of \mathbb{R}^d , from a finite collection of samples X , we rely on a *simplicial complex* K , a discrete structure built over the observed points to provide a topological approximation of the underlying space. Two common examples are the Čech complex and the Vietoris-Rips complex. The *Čech complex* is the simplicial complex where k -simplices correspond to the nonempty intersection of $k + 1$ balls centered at vertices. The *Vietoris-Rips* (or simply *Rips*) *complex* is the simplicial complex where simplexes are built based on pairwise distances among its vertices. We refer to Appendix A for formal definitions.

A collection of simplicial complexes $\mathcal{F} = \{K_a \subset K : a \in \mathbb{R}\}$ satisfying $K_a \subset K_b$ whenever $a \leq b$ is called a *filtration* of K . A typical way of setting the filtration is through a monotonic function on the simplex. A function $f : K \rightarrow \mathbb{R}$ is monotonic if $f(\varsigma) \leq f(\tau)$ whenever ς is a face of τ . If we let $K_a := f^{-1}(-\infty, a]$, then the monotonicity implies that K_a is a subcomplex of K and $K_a \subset K_b$ whenever $a \leq b$. In this paper, we assume that the filtration is built upon a monotonic function.

Persistent homology is a multiscale approach to represent the topological features of the complex K , and can be represented in the persistence diagram. For a filtration \mathcal{F} and for each nonnegative k , we keep track of when k -dimensional homological features (e.g., 0-dimension: connected component, 1-dimension: loop, 2-dimension: cavity, ...) appear and disappear in the filtration. If a homological feature α_i appears at b_i and disappears at d_i , then we say α_i is born at b_i and dies at d_i . By considering these pairs (b_i, d_i) as points in the plane, one obtains the *persistence diagram* defined as follows.

Definition 2.1 Let $\mathbb{R}_*^2 := \{(b, d) \in (\mathbb{R} \cup \infty)^2 : d > b\}$. A *persistence diagram* \mathcal{D} is a finite multiset of $\{p : p \in \mathbb{R}_*^2\}$. We let \mathbb{D} denote the set of all such \mathcal{D} 's.

We will use $\mathcal{D}_X, \mathcal{D}_{\mathbb{X}}$ as shorthand notations for the persistence diagram drawn from the simplicial complex constructed on original data source X, \mathbb{X} , respectively.

Lastly, we define the following metrics to measure the distance between two persistence diagrams.

Definition 2.2 (Bottleneck and Wasserstein distance) Given two persistence diagrams \mathcal{D} and \mathcal{D}' , their *bottleneck distance* (d_B) and q -th *Wasserstein distance* (W_q) for $q \geq 1$ are defined by

$$d_B(\mathcal{D}, \mathcal{D}') = \inf_{\gamma \in \Gamma} \sup_{p \in \mathcal{D}} \|p - \gamma(p)\|_{\infty}, \quad W_q(\mathcal{D}, \mathcal{D}') = \left[\inf_{\gamma \in \Gamma} \sum_{p \in \mathcal{D}} \|p - \gamma(p)\|_{\infty}^q \right]^{\frac{1}{q}}, \quad (1)$$

respectively, where $\|\cdot\|_{\infty}$ is the usual L_{∞} -norm, $\bar{\mathcal{D}} = \mathcal{D} \cup \text{Diag}$ and $\bar{\mathcal{D}}' = \mathcal{D}' \cup \text{Diag}$ with *Diag* being the diagonal $\{(x, x) : x \in \mathbb{R}\} \subset \mathbb{R}^2$ with infinite multiplicity, and the set Γ consists of all the bijections $\gamma : \mathcal{D} \rightarrow \mathcal{D}'$.

Note that for all $q \in [1, \infty)$, $d_B(\mathcal{D}_X, \mathcal{D}_Y) \leq W_q(\mathcal{D}_X, \mathcal{D}_Y)$ for any given $\mathcal{D}_X, \mathcal{D}_Y$. As q tends to infinity, the Wasserstein distance approaches the bottleneck distance. Also, see Appendix B for a further relationship between the bottleneck distance and Wasserstein distance.

2.2 Persistence landscapes

A persistence diagram is a multiset, which is difficult to be used as inputs for machine learning methods (due to the complicated space structure, cardinality issues, computationally inefficient metrics, etc.). Hence, it is useful to transform the persistent homology into a functional Hilbert space, where the analysis is easier and learning methods can be directly applied. One good example is the persistence landscape [Bubenik, 2015, 2018, Bubenik and Dłotko, 2017]. Let \mathcal{D} denote a persistence

diagram that contains N off-diagonal birth-death pairs. We first consider a set of piecewise-linear functions $\{\Lambda_p(t)\}_{p \in \mathcal{D}}$ for all birth-death pairs $p = (b, d) \in \mathcal{D}$ as

$$\Lambda_p(t) = \max\{0, \min\{t - b, d - t\}\}.$$

Then the *persistence landscape* λ of the persistence diagram \mathcal{D} is defined as a sequence of functions $\{\lambda_k\}_{k \in \mathbb{N}}$, where

$$\lambda_k(t) = k \max_p \Lambda_p(t), \quad t \in \mathbb{R}, k \in \mathbb{N}, \quad (2)$$

Hence, the persistence landscape is a set of real-valued functions and is easily computable. Advantages for this kind of functional summaries are discussed in Chazal et al. [2014b], Berry et al. [2018].

2.3 Distance to measure (DTM) function

The Distance to measure (DTM) [Chazal et al., 2011, 2016a] is a robustified version of the distance function. More precisely, the DTM $d_{\mu, m_0} : \mathbb{R}^d \rightarrow \mathbb{R}$ for a probability distribution μ with parameter $m_0 \in (0, 1)$ and $r \geq 1$ is defined by

$$d_{\mu, m_0}(x) = \left(\frac{1}{m_0} \int_0^{m_0} (\delta_{\mu, m}(x))^r dm \right)^{1/r},$$

where $\delta_{\mu, m}(x) = \inf\{t > 0 : \mu(\mathbb{B}(x, t)) > m\}$ when $\mathbb{B}(x, t)$ is an open ball centered at x with radius t . If not specified, $r = 2$ is used as a default. In practice, we use a weighted empirical measure

$$P_n(x) = \frac{\sum_{i=1}^n \varpi_i \mathbb{1}(X_i = x)}{\sum_{i=1}^n \varpi_i},$$

with weights ϖ_i 's for μ . In this case, we define the *empirical DTM* by

$$\hat{d}_{m_0}(x) = d_{P_n, m_0}(x) = \left(\frac{\sum_{X_i \in N_k(x)} \varpi'_i \|X_i - x\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{1/r}, \quad (3)$$

where $N_k(x)$ is the subset of $\{X_1, \dots, X_n\}$ containing the k nearest neighbors of x , k is such that $\sum_{X_i \in N_{k-1}(x)} \varpi_i < m_0 \sum_{i=1}^n \varpi_i \leq \sum_{X_i \in N_k(x)} \varpi_i$, and $\varpi'_i = \sum_{X_j \in N_k(x)} \varpi_j - m_0 \sum_{j=1}^n \varpi_j$ if at least one of X_i 's is in $N_k(x)$ and $\varpi'_i = \varpi_i$ otherwise. Hence the empirical DTM behaves similarly to the k -nearest distance with $k = \lfloor m_0 n \rfloor$. For i.i.d cases, we typically set $\varpi_i = 1$ but the weights can be flexibly determined in data-driven way. The parameter m_0 determines how much topological/geometrical information should be extracted from the local or global structure. A brief guideline on DTM parameter selection can be found in Appendix F (see Chazal et al. [2011] for more details). Since the resulting persistence diagram is less prone to input perturbations and has nice stability properties, people often prefer using the DTM as their filtration function.

3 A novel topological layer based on weighted persistence landscapes

In this section, we present a detailed algorithm to implement PLLay for a general neural network. Let X , \mathcal{D}_X , h_{top} denote our input, corresponding persistence diagram induced from X , the proposed topological layer, respectively. Broadly speaking, the construction of our proposed topological layer consists of two steps: 1) computing a persistence diagram from the input, and 2) constructing the topological layer from the persistence diagram.

3.1 Computation of diagram: $X \rightarrow \mathcal{D}_X$

To compute the persistence diagram from the input data, we first need to define the filtration which requires a simplicial complex K and a function $f : K \rightarrow \mathbb{R}$. There are several options for K and f . We are in general agnostic about which filtration to use since it is in fact problem-dependent; in practice, we suggest using ensemble-like methods that can adapt to various underlying topological structures. One popular choice is the Vietoris-Rips filtration. When there is a one-to-one correspondence between X_i and each fixed grid point Y_i , one obvious choice for f could be just interpreting X as a function values, so $f(Y_i) = X_i$. We refer to Chazal and Michel [2017] for more examples.

As described in Section 2.3, one appealing choice for f is the DTM function. Due to its favorable properties, the DTM function has been widely used in TDA [Anai et al., 2019, Xu et al., 2019], and

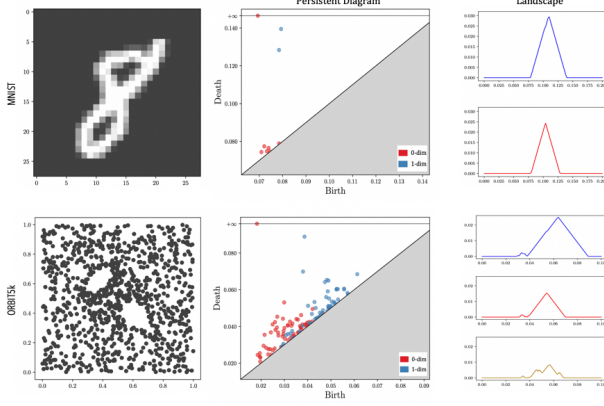


Figure 2: The topological features encoded in the persistence diagram & persistence landscapes for MNIST and ORBIT5k sample. In the MNIST example, two loops (1-dimensional feature) in ‘8’ are clearly identified and encoded into the 1st and 2nd order landscapes. The ORBIT5k sample shows more involved patterns.

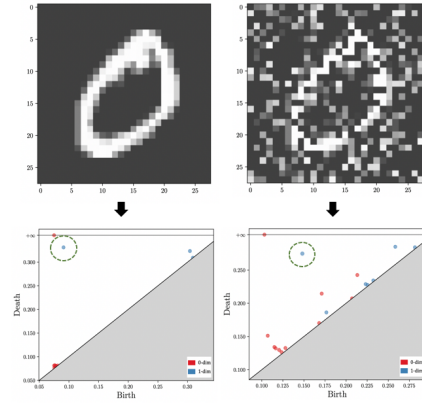


Figure 3: The significant point (inside green-dashed circle) in the persistence diagram remains almost unchanged even after corrupting pixels and adding noise to the image.

has a good potential for deep learning application. Nonetheless, to the best of our knowledge, the DTM function has not yet been adopted in previous studies. In what follows, we detail two common scenarios for the DTM adaptation: when we consider the input X as 1) data points or 2) weights.

- If the input data X is considered as the empirical data points, then the empirical DTM in (3) with weights ϖ_i ’s becomes
- If the input data X is considered as the weights corresponding to fixed points $\{Y_1, \dots, Y_n\}$, then the empirical DTM in (3) with data points Y_i ’s and weights X_i ’s becomes

$$\hat{d}_{m_0}(x) = \left(\frac{\sum_{X_i \in N_k(x)} \varpi'_i \|X_i - x\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{1/r}, \quad (4)$$

where k and ϖ'_i are determined as in (3).

$$\hat{d}_{m_0}(x) = \left(\frac{\sum_{X_i \in N_k(x)} X'_i \|Y_i - x\|^r}{m_0 \sum_{i=1}^n X_i} \right)^{1/r}, \quad (5)$$

where k and ϖ'_i are determined as in (3).

Figure 2 provides some real data examples (which will be used in Section 5) of the persistence diagrams and the corresponding persistence landscapes based on the DTM functions. As shown in Figure 3, the topological features are expected to be robust to external noise or corruption.

3.2 Construction of topological layer: $\mathcal{D}_X \rightarrow h_{\text{top}}$

Our topological layer is defined based on a parametrized mapping which takes the persistence diagram \mathcal{D} to be projected onto \mathbb{R} , by harnessing persistence landscapes. Our construction is less afflicted by the artificial bending due to a particular transformation procedure as in Hofer et al. [2017], yet still guarantees the crucial information in the persistence diagram to be well preserved as will be seen in Section 4. Insignificant points with low persistence are likely to be ignored systematically without introducing additional nuisance parameters [Bubenik and Dłotko, 2017].

Let \mathbb{R}^{+0} denote $[0, \infty)$. Given a persistence diagram $\mathcal{D} \in \mathbb{D}$, we compute the persistence landscape of order k in (2), $\lambda_k(t)$, for $k = 1, \dots, K_{\max}$. Then, we compute the weighted average $\bar{\lambda}_\omega(t) := \sum_{k=1}^{K_{\max}} \omega_k \lambda_k(t)$ with a weight parameter $\omega = \{\omega_k\}_k$, $\omega_k > 0$, $\sum_k \omega_k = 1$. Next, we set a domain $[T_{\min}, T_{\max}]$ and a resolution $\nu := T/(m-1)$, and sample m equal-interval points from $[T_{\min}, T_{\max}]$ to obtain $\bar{\Lambda}_\omega = (\bar{\lambda}_\omega(T_{\min}), \bar{\lambda}_\omega(T_{\min} + \nu), \dots, \bar{\lambda}_\omega(T_{\max}))^\top \in (\mathbb{R}^{+0})^m$. Consequently, we have defined a mapping $\bar{\Lambda}_\omega: \mathbb{D} \rightarrow (\mathbb{R}^{+0})^m$ which is a (vectorized) finite-sample approximation of the weighted persistence landscapes at the resolution ν , at fixed, predetermined locations. Finally, we consider a parametrized differentiable map $g_\theta: (\mathbb{R}^{+0})^m \rightarrow \mathbb{R}$ which takes the input $\bar{\Lambda}_\omega$ and is differentiable with respect to θ as well. Now, the projection of \mathcal{D} with respect to the mapping

Algorithm 1 Implementation of single structure element for PLLay

Input: persistence diagram $\mathcal{D} \in \mathbb{D}$

1. compute $\lambda_k(t)$ (2) on $t \in [0, T]$ for every $k = 1, \dots, K_{max}$
2. compute the weighted average $\bar{\lambda}_\omega(t) := \sum_{k=1}^{K_{max}} \omega_k \lambda_k(t)$, $\omega_k > 0$, $\sum_k \omega_k = 1$
3. set $\nu := \frac{T}{m-1}$, and compute $\bar{\Lambda}_\omega = (\bar{\lambda}_\omega(T_{\min}), \bar{\lambda}_\omega(T_{\min} + \nu), \dots, \bar{\lambda}_\omega(T_{\max}))^\top \in \mathbb{R}^m$
4. for a parametrized differentiable map $g_\theta: \mathbb{R}^m \rightarrow \mathbb{R}$, define $S_{\theta, \omega} = g_\theta \circ \bar{\Lambda}_\omega$

Output: $S_{\theta, \omega}: \mathbb{D} \rightarrow \mathbb{R}$

$S_{\theta, \omega} := g_\theta \circ \bar{\Lambda}_\omega$ defines a single *structure element* for our topological input layer. We summarize the procedure in Algorithm 1.

The projection $S_{\theta, \omega}$ is continuous at every $t \in [T_{\min}, T_{\max}]$. Also, note that it is differentiable with respect to ω and θ , regardless of the resolution level ν . In what follows, we provide some guidelines that might be useful to implement Algorithm 1.

ω : The weight parameter ω can be initialized uniformly, i.e. $\omega_k = 1/K_{max}$ for all k , and will be re-determined during training through the softmax layer in a way that a certain landscape conveying significant information has more weight. In general, lower-order landscapes tend to be more significant than higher-order landscapes, but the optimal weights may vary from task to task.

θ, g_θ : Likewise, some birth-death pairs, encoded in the landscape function, may contain more crucial information about the topological features of the input data structure than others. Roughly speaking, this is equivalent to say certain mountains (or their ridge or valley) in the landscape are especially important. Hence, the parametrized map g_θ should be able to reflect this by its design. In general, it can be done by affine transformation with scale and translation parameter, followed by an extra nonlinearity and normalization if necessary. We list two possible choices as below.

- Affine transformation: with scale and translation parameter $\sigma_i, \mu_i \in \mathbb{R}^m$, $g_{\theta_i}(\bar{\Lambda}_\omega) = \sigma_i^\top (\bar{\Lambda}_\omega - \mu_i)$ and $\theta_i = (\sigma_i, \mu_i)$.
- Logarithmic transformation: with same $\theta_i = (\sigma_i, \mu_i)$, $g_{\theta_i}(\bar{\Lambda}_\omega) = \exp(-\sigma_i \|\bar{\Lambda}_\omega - \mu_i\|_2)$.

Note that other constructions of g_θ, θ, ω are also possible as long as they satisfy the sufficient conditions described above. Finally, since each structure element corresponds to a single node in a layer, we concatenate many of them, each with different parameters, to form our topological layer.

Definition 3.1 (Persistence landscape-based topological layer (PLLay)) For $n_h \in \mathbb{N}$, let $\eta_i = (\theta_i, \omega_i)$ denote the set of parameters for the i -th structure element and let $\eta = (\eta_i)_{i=1}^{n_h}$. Given \mathcal{D} and resolution ν , we define PLLay as a parametrized mapping with η of $\mathbb{D} \rightarrow \mathbb{R}^{n_h}$ such that

$$h_{top}: \mathcal{D} \rightarrow (S_{\eta_i}(\mathcal{D}; \nu))_{i=1}^{n_h}. \quad (6)$$

Note that this is nothing but a concatenation of n_h topological structure elements (nodes) with different parameter sets (thus n_h is our layer dimension).

Remark 1 Our PLLay considers only K_{max} top landscape functions. For a given persistence diagram, the points near the diagonal are not likely to appear at K_{max} top landscape functions, and hence not considered in PLLay. And hence PLLay automatically filters out the noisy features.

3.3 Differentiability

This subsection is devoted to the analysis of the differential behavior of PLLay with respect to its input (or output from the previous layer), by computing the derivatives $\frac{\partial h_{top}}{\partial X}$. Since $\frac{\partial h_{top}}{\partial X} = \frac{\partial h_{top}}{\partial \mathcal{D}_X} \circ \frac{\partial \mathcal{D}_X}{\partial X}$, this can be done by combining two derivatives $\frac{\partial \mathcal{D}_X}{\partial X}$ and $\frac{\partial h_{top}}{\partial \mathcal{D}_X}$. We have extended Poulenard et al. [2018] so that we can compute the above derivatives for general persistent homology under arbitrary filtration in our setting. We present the result in Theorem 3.1.

Theorem 3.1 Let f be the filtration function. Let ξ be a map from each birth-death point $(b_i, d_i) \in \mathcal{D}_X$ to a pair of simplices (β_i, δ_i) . Suppose that ξ is locally constant at X , and $f(\beta_i)$ and $f(\delta_i)$ are differentiable with respect to X_j 's. Then, h_{top} is differentiable with respect to X and

$$\frac{\partial h_{top}}{\partial X_j} = \sum_i \frac{\partial f(\beta_i)}{\partial X_j} \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial b_i} + \sum_i \frac{\partial f(\delta_i)}{\partial X_j} \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial d_i}.$$

The proof is in Appendix E.1. Note that $\frac{\partial \lambda_k}{\partial b_i}, \frac{\partial \lambda_k}{\partial d_i}$ are piecewise constant and are easily computed in explicit forms. Also $\frac{\partial g_{\theta}}{\partial x_l}$ can be easily realized by an automatic differentiation framework such as tensorflow or pytorch. Our PLLay in Definition 3.1 is thus trainable via backpropagation at an arbitrary location in the network. In Appendix D, we also provide a derivative for the DTM filtration.

4 Stability Analysis

A key property of PLLay is stability; its discriminating power should remain stable against non-systematic noise or perturbation of input data. In this section, we shall provide our theoretical results on the stability properties of the proposed layer. We first address the stability for each structure element with respect to changes in persistence diagrams in Theorem 4.1.

Theorem 4.1 Let g_{θ} be $\|\cdot\|_{\infty}$ -Lipschitz, i.e. there exists $L_g > 0$ with $|g_{\theta}(x) - g_{\theta}(y)| \leq L_g \|x - y\|_{\infty}$ for all $x, y \in \mathbb{R}^m$. Then for two persistence diagrams $\mathcal{D}, \mathcal{D}'$,

$$|S_{\theta, \omega}(\mathcal{D}; \nu) - S_{\theta, \omega}(\mathcal{D}'; \nu)| \leq L_g d_B(\mathcal{D}, \mathcal{D}').$$

Proof of Theorem 4.1 is given in Appendix E.2. Theorem 4.1 shows that $S_{\theta, \omega}$ is stable with respect to perturbations in the persistence diagram measured by the bottleneck distance (1). It should be noted that only the Lipschitz continuity of g_{θ} is required to establish the result.

Next, Corollary 4.1 shows that under certain conditions our approach improves the previous stability result of Hofer et al. [2017].

Corollary 4.1 For $t > 0$, let $n_t \in \mathbb{N}$ be satisfying that, for any two diagrams $\mathcal{D}_t, \mathcal{D}'_t$ with $d_B(\mathcal{D}, \mathcal{D}_t) \leq t$ and $d_B(\mathcal{D}', \mathcal{D}'_t) \leq t$, either $\mathcal{D}_t \setminus \mathcal{D}'_t$ or $\mathcal{D}'_t \setminus \mathcal{D}_t$ has at least n_t points. Then, the ratio of our stability bound in Theorem 4.1 to that in Hofer et al. [2017] is upper bounded by

$$C_{g_{\theta}} / (1 + (2t/d_B(\mathcal{D}, \mathcal{D}')) \times (n_t - 1)),$$

where $C_{g_{\theta}}$ is a constant to be specified in the proof.

See Appendix E.3 for the proof. Corollary 4.1 implies that for complex data structures where each \mathcal{D} contains many birth-death pairs (for fixed t , in general n_t grows with the increase in the number of points in \mathcal{D}), our stability bound is tighter than that of Hofer et al. [2017] at polynomial rates.

In particular, when we use the DTM function-based filtration proposed in (4) and (5), Theorem 4.1 can be turned into the following stability result with respect to our input X .

Theorem 4.2 Suppose $r = 2$ is used for the DTM function. Let a differentiable function g_{θ} and resolution ν be given, and let P be a distribution. For the case when X_j 's are data points, i.e. when (4) is used as the DTM function of X , let P_n be the empirical distribution defined by $P_n = \frac{\sum_{i=1}^n \varpi_i \delta_{X_i}}{\sum_{i=1}^n \varpi_i}$. For the case when X_j 's are weights, i.e. when (5) is used as the DTM function of X , let P_n be the empirical distribution defined by $P_n = \frac{\sum_{i=1}^n X_i \delta_{Y_i}}{\sum_{i=1}^n X_i}$. Let \mathcal{D}_P be the persistence diagram of the DTM filtration of P , and \mathcal{D}_X be the persistence diagram of the DTM filtration of X . Then,

$$|S_{\theta, \omega}(\mathcal{D}_X; \nu) - S_{\theta, \omega}(\mathcal{D}_P; \nu)| \leq L_g m_0^{-1/2} W_2(P_n, P).$$

The proof is given in Appendix E.4. Theorem 4.2 implies that if the empirical distribution P_n induced from the given input X well approximates the true distribution P with respect to the Wasserstein distance, i.e. having small $W_2(P_n, P)$, then PLLay constructed on observed data is close to the one as if we were to know the true distribution P .

Theorem 4.1 and 4.2 suggest that the topological information embedded in the proposed layer is robust against small noise, data corruption, or outliers. We have also discussed the stability result for the Vietoris-Rips and the Čech complex in Appendix C.

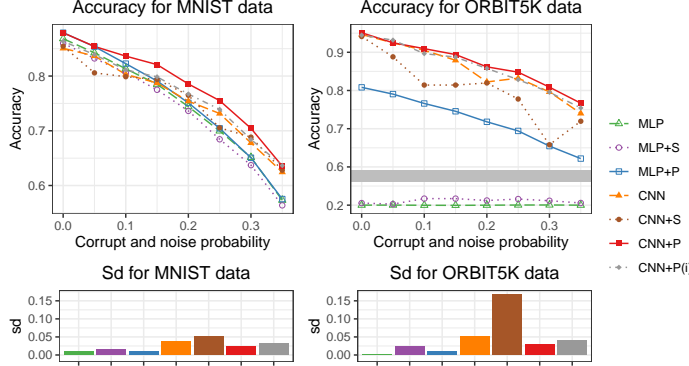


Figure 4: Test accuracy in MNIST and ORBIT5K experiments. PLLay consistently improves the accuracy and the robustness against noise and corruption. In particular, in many cases it effectively reduces the variance of the classification accuracy on ORBIT5K.

Model	Accuracy
PointNet	0.708 (± 0.285)
PersLay	0.877 (± 0.010)
CNN	0.915 (± 0.088)
CNN+	0.943 (± 0.014)
SLay	0.950 (± 0.014)
CNN+	0.950 (± 0.016)
PL Lay	0.950 (± 0.016)

Table 1: Comparison of different methods for ORBIT5K including the current state-of-the-art PersLay. The proposed method achieves the new state-of-the-art accuracy.

5 Experiments

To demonstrate the effectiveness of the proposed approach, we study classification problems on two different datasets: MNIST handwritten digits and ORBIT5K. To fairly showcase the benefits of using our proposed method, we keep our network architecture as simple as possible so that we can focus on the contribution from PLLay. In the experiments, we aim to explore the benefits of our layer through the following questions: 1) does it make the network more robust and reliable against noise, etc.? and 2) does it improve the overall generalization capability compared to vanilla models? In order to address both of these questions, we first consider the *corruption process*, a certain amount of random omission of pixel values or points from each raw example (so we will have less information), and the *noise process*, a certain amount of random addition of uniformly-distributed noise signals or points to each raw example. An example is given in Figure 3. Then we fit a standard multilayer perceptron (MLP) and a convolutional neural network (CNN) with and without the augmentation of PLLay across various noise and corruption rates given to the raw data, and compare the results. The guideline for choosing the TDA parameters in this experiment is described in Appendix F. We intentionally use a small number of training data (~ 1000) so that the convergence rates could be included in the evaluation criteria. Each simulation is repeated 20 times. We refer to Appendix G for details about each simulation setup and our model architectures.

MNIST handwritten digits

We classify handwritten digit images from MNIST dataset. Each digit has distinctive topological information which can be encoded into the Persistence Landscape as in Figure 2.

Topological layer. We add two parallel PLLays in Definition 6 at the beginning of MLP and CNN models, based on the empirical DTM function in (5), where we define fixed 28×28 points on grid and use a set of grayscale values X as a weight vector for the fixed points. We used $m_0 = 0.05$ and $m_0 = 0.2$ for each layer, respectively (referred to MLP+P, CNN+P(i), respectively). Particularly for the CNN model, it is likely that the output of the convolutional layers might carry significant information about (smoothed) geometry of the input data shape. So we additionally place another PLLay after each convolutional layer, directly taking the layer output as 2D-function values and using the sublevel filtration (CNN+P).

Baselines. As our baseline methods, we employ 2-layer vanilla MLP, 2-layer CNN, and the topological signature method by Hofer et al. [2017] based on the empirical DTM function proposed in (5) (which we will refer to as SLay). The SLay is augmented at the beginning of MLP and CNN, referred to as MLP+S and CNN+S. See Appendix G.1 for more details.

Result. In Figure 4, we observe that PLLay augmentation consistently improves the accuracy of all the baselines. Interestingly, as we increase the corruption and noise rates, the improvement on

CNN increases up to the moderate level of corruption and noise ($\sim 15\%$), then starts to decrease. We conjecture that this is because although DTM filtration is able to robustly capture homological signals as illustrated in Figure 2, if the corruption and noise levels become too much, then the topological structure starts to dissolve in the DTM filtration.

Orbit Recognition

We classify point clouds generated by 5 different dynamical systems from ORBIT5K dataset [Adams et al., 2017, Carrière et al., 2020]. The detailed data generating process is described in Appendix G.2.

Topological layer. The setup remains the same as in the previous MNIST case, except that 1) PLLay at the beginning of each network uses the empirical DTM function in (4), and 2) we set $m_0 = 0.02$.

Baselines & Simulation. All the baseline methods remain the same. For noiseless case, we added PointNet [Charles et al., 2017], a state-of-the-art in point cloud classification, and PersLay [Carrière et al., 2020], a state-of-the-art in TDA-utilized classification.

Result. In Figure 4, we observe that PLLay improves upon MLP and MLP+S by a huge margin ($42\% \sim 60\%$). In particular, without augmenting PLLay, MLP and MLP+S remain at almost a random classifier, which implies that the topological information is indeed crucial for the ORBIT5K classification task, and it would otherwise be very challenging to extract meaningful features. PLLay improves upon CNN or CNN+S consistently as well. Moreover, it appears that CNN suffers from high variance due to the high complexity of ORBIT5K dataset. On the other hand, PLLay can effectively mitigate this problem and make the model more stable by utilizing robust topological information from DTM function. Impressively, for the noiseless case, PLLay has achieved better performance than all the others including the current state-of-the-art PointNet and PersLay by a large margin.

6 Discussion

In this study, we have presented PLLay, a novel topological layer based on the weighted persistence landscape where we can exploit the topological features effectively. We provide the differentiability guarantee of the proposed layer with respect to the layer’s input under arbitrary filtration. Hence, our study offers the first general topological layer which can be placed anywhere in the deep learning network. We also present new stability results that verify the robustness and efficiency of our approach. It is worth noting that our method and analytical results in this paper can be extended to silhouettes [Chazal et al., 2015, 2014b]. In the experiments, we have achieved the new state-of-the-art accuracy for ORBIT5K dataset based on the proposed method. We expect our work to bridge the gap between modern TDA tools and deep learning research.

The computational complexity depends on how PLLay is used. Computing the DTM is $O(n+m \log n)$ when $m_0 \propto 1/n$ and k-d tree is used, where n is the input size and m is the grid size. Computing the persistence diagram is $O(m^{2+\epsilon})$ for any small $\epsilon > 0$ when the simplicial complex K in Section 3.1 grows linearly with respect to the grid size such as cubical complex or alpha complex (Chen and Kerber [2013] and Theorem 4.4, 5.6 of Boissonnat et al. [2018]). Computing the persistence landscape grows linearly with respect to the number of homological features in the persistence diagram, which is the topological complexity of the input and does not necessarily depend on n or m . For our experiments, we consider fixed grids of size 28×28 and 40×40 as in Appendix G, so the computation is not heavy. Also, if we put PLLay only at the beginning of the deep learning model, then PLLay can be pre-computed and needs not to be calculated at every epoch in the training.

There are several remarks regarding our experiments. First, we emphasize that SLay in Section 5 is rather an intermediate tool designed for our simulation and not completely identical to the topological signature method by Hofer et al. [2017]. For example, SLay combines the method by Hofer et al. [2017] and the DTM function in (4) and (5) that have not appeared in the previous study. So we cannot exclude the possibility that the comparable performance of SLay for certain simulations is due to the contribution by the DTM function filtration. Moreover, for CNN, placing extra PLLay after each convolutional layer appears to bring marginal improvement in accuracy in our experiments. Exploring the optimal architecture with our PLLay, e.g., finding the most accurate and efficient PLLay network for a given classification task, would be an interesting future work.

The source code of PLLay is publicly available at <https://github.com/jisuk1/pllay/>.

Broader Impact

This paper proposes a novel method of adapting tools in applied mathematics to enhance the learnability of deep learning models. Even though our methodology is generally applicable to any complex modern data, it is not tuned to a specific application that might improperly incur direct societal/ethical consequences. So the broader impact discussion is not needed for our work.

Acknowledgments and Disclosure of Funding

During the last 36 months prior to the submission, Jisu Kim received Samsung Scholarship, and Joon Sik Kim received Kwanjeong Fellowship. Frédéric Chazal was supported by the ANR AI chair TopAI.

References

- Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: a stable vector representation of persistent homology. *J. Mach. Learn. Res.*, 18:Paper No. 8, 35, 2017. ISSN 1532-4435.
- Saleh Albelwi and Ausif Mahmood. Automated optimal architecture of deep convolutional neural networks for image recognition. In *2016 15th IEEE International conference on machine learning and applications (ICMLA)*, pages 53–60. IEEE, 2016.
- Hirokazu Anai, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hiroya Inakoshi, Raphaël Tinarrage, and Yuhei Umeda. Dtm-based filtrations. In *35th International Symposium on Computational Geometry (SoCG 2019)*, 2019.
- S. A. Barannikov. The framed Morse complex and its invariants. In *Singularities and bifurcations*, volume 21 of *Adv. Soviet Math.*, pages 93–115. Amer. Math. Soc., Providence, RI, 1994.
- Eric Berry, Yen-Chi Chen, Jessi Cisewski-Kehe, and Brittany Terese Fasy. Functional summaries of persistence diagrams. *arXiv preprint arXiv:1804.01618*, 2018.
- Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and topological inference*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2018. ISBN 978-1-108-41089-2; 978-1-108-41939-0. doi: 10.1017/9781108297806. URL <https://doi.org/10.1017/9781108297806>.
- Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.
- Peter Bubenik. The persistence landscape and some of its properties. *arXiv preprint arXiv:1810.04963*, 2018.
- Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.
- Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A course in metric geometry*, volume 33 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001. ISBN 0-8218-2129-6. doi: 10.1090/gsm/033. URL <https://doi.org/10.1090/gsm/033>.
- Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule, editors, *Topological Data Analysis*, pages 119–146, Cham, 2020. Springer International Publishing. ISBN 978-3-030-43408-3.
- Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.

- Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2786–2796, Online, 26–28 Aug 2020. PMLR. URL <http://proceedings.mlr.press/v108/carriere20a.html>.
- R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.
- Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.
- Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246. ACM, 2009.
- Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.
- Frédéric Chazal, Vin De Silva, and Steve Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, 2014a.
- Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes and silhouettes. In *Proceedings of the thirtieth annual symposium on Computational geometry*, page 474. ACM, 2014b.
- Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Subsampling methods for persistent homology. In *International Conference on Machine Learning*, pages 2143–2151, 2015.
- Frédéric Chazal, Pascal Massart, Bertrand Michel, et al. Rates of convergence for robust geometric inference. *Electronic journal of statistics*, 10(2):2243–2286, 2016a.
- Frédéric Chazal, Steve Y. Oudot, Marc Glisse, and Vin De Silva. *The Structure and Stability of Persistence Modules*. SpringerBriefs in Mathematics. Springer Verlag, 2016b. URL <https://hal.inria.fr/hal-01330678>.
- Chao Chen and Michael Kerber. An output-sensitive algorithm for persistent homology. *Comput. Geom.*, 46(4):435–447, 2013. ISSN 0925-7721. doi: 10.1016/j.comgeo.2012.02.010. URL <https://doi.org/10.1016/j.comgeo.2012.02.010>.
- Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7:339–358, 2007. ISSN 1472-2747. doi: 10.2140/agt.2007.7.339. URL <https://doi.org/10.2140/agt.2007.7.339>.
- Meryll Dindin, Yuhei Umeda, and Frédéric Chazal. Topological data analysis for arrhythmia detection through modular neural networks. In *Canadian Conference on Artificial Intelligence*, pages 177–188. Springer, 2020.
- Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.
- Saba Emrani, Thanos Gentimis, and Hamid Krim. Persistent homology of delay embeddings and its application to wheeze detection. *IEEE Signal Processing Letters*, 21(4):459–463, 2014.
- Brittany T. Fasy, Jisu Kim, Fabrizio Lecci, Clément Maria, David L. Millman, and Vincent Rouvreau. Introduction to the R package TDA. *CoRR*, abs/1411.1830, 2014. URL <http://arxiv.org/abs/1411.1830>.

- Rickard Br  el Gabri  lsson, Bradley J. Nelson, Anjan Dwaraknath, Primo   Skraba, Leonidas J. Guibas, and Gunnar E. Carlsson. A topology layer for machine learning. *CoRR*, 2019. URL <http://arxiv.org/abs/1905.12200>.
- Jennifer Gamble and Giseon Heo. Exploring uses of persistent homology for statistical analysis of landmark-based shape data. *Journal of Multivariate Analysis*, 101(9):2184–2199, 2010.
- Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.
- William H Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *arXiv preprint arXiv:1802.04443*, 2018.
- Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1634–1644, 2017.
- Christoph Hofer, Roland Kwitt, Mandar Dixit, and Marc Niethammer. Connectivity-optimized representation learning via persistent homology. *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- Y LeCun, L Bottou, Y Bengio, et al. Lenet-5, convolutional neural networks (2015). Retrieved June, 1, 2016.
- Jen-Yu Liu, Shyh-Kang Jeng, and Yi-Hsuan Yang. Applying topological persistence in convolutional neural network for music audio signals. *arXiv preprint arXiv:1608.07373*, 2016.
- Michael Moor, Max Horn, Bastian Rieck, and Karsten M. Borgwardt. Topological autoencoders. *CoRR*, abs/1906.00722, 2020. URL <http://arxiv.org/abs/1906.00722>.
- Vidit Nanda and Radmila Sazdanovi  . Simplicial models and topological inference in biological systems. In *Discrete and topological models in molecular biology*, pages 109–141. Springer, 2014.
- C  ssio MM Pereira and Rodrigo F de Mello. Persistent homology for time series and spatial data clustering. *Expert Systems with Applications*, 42(15-16):6026–6038, 2015.
- Adrien Poulenard, Primo   Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. *Computer Graphics Forum*, 37(5):13–25, 2018. doi: 10.1111/cgf.13487. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13487>.
- Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ByxkijC5FQ>.
- Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

- The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.3.0 edition, 2020. URL <https://gudhi.inria.fr/doc/3.3.0/>.
- Christopher J Tralie and Jose A Perea. (quasi) periodicity quantification in video data, using topology. *SIAM Journal on Imaging Sciences*, 11(2):1049–1077, 2018.
- Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.
- Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan Turaga. Persistent homology of attractors for action recognition. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 4150–4154. IEEE, 2016.
- X. Xu, J. Cisewski-Kehe, S. B. Green, and D. Nagai. Finding cosmic voids and filament loops using topological data analysis. *Astronomy and Computing*, 27:34, Apr 2019. doi: 10.1016/j.ascom.2019.02.003.
- Xiaojin Zhu. Persistent homology: An introduction and a new text representation for natural language processing. In *IJCAI*, pages 1953–1959, 2013.
- Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

APPENDIX

A Simplicial complex, Persistent homology, and Distance between sets on metric spaces

Throughout, we will let \mathbb{X} denotes a subset of \mathbb{R}^d , and X denotes a finite collection of points from an arbitrary space \mathbb{X} .

A simplicial complex can be seen as a high dimensional generalization of a graph. Given a set V , an (abstract) *simplicial complex* is a set K of finite subsets of V such that $\alpha \in K$ and $\beta \subset \alpha$ implies $\beta \in K$. Each set $\alpha \in K$ is called its *simplex*. The *dimension* of a simplex α is $\dim \alpha = \text{card} \alpha - 1$, and the dimension of the simplicial complex is the maximum dimension of any of its simplices. Note that a simplicial complex of dimension 1 is a graph.

When approximating the topology of the underlying space by observed samples, a common choice is the *Čech complex*, defined next. Below, for any $x \in \mathbb{X}$ and $r > 0$, we let $\mathbb{B}_{\mathbb{X}}(x, r)$ denote the open ball centered at x and radius $r > 0$ intersected with \mathbb{X} .

Definition A.1 (Čech complex) *Let $\mathcal{X} \subset \mathbb{X}$ be finite and $r > 0$. The (weighted) Čech complex is the simplicial complex*

$$\check{\text{Cech}}_{\mathcal{X}}^{\mathbb{X}}(r) := \{\sigma \subset \mathcal{X} : \cap_{x \in \sigma} \mathbb{B}_{\mathbb{X}}(x, r) \neq \emptyset\}. \quad (7)$$

The superscript \mathbb{X} will be dropped when understood from the context.

Another common choice is the *Vietoris-Rips complex*, also referred to as *Rips complex*, where simplexes are built based on pairwise distances among its vertices.

Definition A.2 (Vietoris-Rips complex) *Let $\mathcal{X} \subset \mathbb{X}$ be finite and $r > 0$. The Vietoris-Rips complex $\text{Rips}_{\mathcal{X}}(r)$ is the simplicial complex defined as*

$$\text{Rips}_{\mathcal{X}}(r) := \{\sigma \subset \mathcal{X} : d(x_i, x_j) < 2r, \forall x_i, x_j \in \sigma\}. \quad (8)$$

Note that from (7) and (8), the Čech complex and Vietoris-Rips complex have the following interleaving inclusion relationship

$$\check{\text{Cech}}_{\mathcal{X}}(r) \subset \text{Rips}_{\mathcal{X}}(r) \subset \check{\text{Cech}}_{\mathcal{X}}(2r).$$

In particular, when $\mathbb{X} \subset \mathbb{R}^d$ is a subset of a Euclidean space of dimension d , then the constant 2 can be tightened to $\sqrt{\frac{2d}{d+1}}$ (e.g., see Theorem 2.5 in de Silva and Ghrist [2007]):

$$\check{\text{Cech}}_{\mathcal{X}}(r) \subset \text{Rips}_{\mathcal{X}}(r) \subset \check{\text{Cech}}_{\mathcal{X}}\left(\sqrt{\frac{2d}{d+1}}r\right).$$

Persistent homology [Barannikov, 1994, Zomorodian and Carlsson, 2005, Edelsbrunner et al., 2000, Chazal et al., 2014a] is a multiscale approach to represent topological features of the complex K . A *filtration* \mathcal{F} is a collection of subcomplexes approximating the data points at different resolutions, formally defined as follows.

Definition A.3 (Filtration) *A filtration $\mathcal{F} = \{K_a \subset K\}_{a \in \mathbb{R}}$ is a collection of subcomplexes of K such that $a \leq b$ implies that $K_a \subset K_b$.*

For a filtration \mathcal{F} and for each $k \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$, the associated persistent homology $PH_k \mathcal{F}$ is an ordered collection of k -th dimensional homologies, one for each element of \mathcal{F} .

Definition A.4 (Persistent homology) *Let \mathcal{F} be a filtration and let $k \in \mathbb{N}_0$. The associated k -th persistent homology $PH_k \mathcal{F}$ is a collection of groups $\{H_k(K_a)\}_{a \in \mathbb{R}}$ of each subcomplex K_a in \mathcal{F} equipped with homomorphisms $\{v_k^{a,b}\}_{a \leq b}$, where $H_k(K_a)$ is the k -th dimensional homology group of K_a and $v_k^{a,b} : H_k K_a \rightarrow H_k K_b$ is the homomorphism induced by the inclusion $K_a \subset K_b$.*

For the k -th persistent homology $PH_k\mathcal{F}$, the set of filtration levels at which a specific homology appears is always an interval $[b, d) \subset [-\infty, \infty]$, i.e. a specific homology is formed at some filtration value b and dies when the inside hole is filled at another value $d > b$. To be more formally, the image of a specific homology class α in $H_k(K_a)$ is nonzero if and only if $b \leq a < d$. We often say that α is born at b and dies at d . By considering these pairs as points in the plane, one obtains the *persistence diagram* as below.

Definition A.5 (Persistence diagram) Let $\mathbb{R}_*^2 := \{(b, d) \in (\mathbb{R} \cup \infty)^2 : d > b\}$. Let \mathcal{F} be a filtration and let $k \in \mathbb{N}_0$. The corresponding k -th persistence diagram $Dgm_k(\mathcal{F})$ is a finite multiset of \mathbb{R}_*^2 , consisting of all pairs (b, d) , where $[b, d)$ is the interval of filtration values for which a specific homology class appears in $PH_k\mathcal{F}$. b is called a birth time and d is called a death time.

When topological information of the underlying space is approximated by the observed points, it is often needed to compare two sets with respect to their metric structures. Here we present two distances on metric spaces, Hausdorff distance and Gromov-Hausdorff distance. We refer to Burago et al. [2001] for more details and other distances.

The *Hausdorff distance* [Burago et al., 2001, Definition 7.3.1] is on sets embedded in the same metric spaces. This distance measures how two sets are close to each other in the embedded metric space. When $S \subset \mathbb{X}$, we denote by $U_r(S)$ the r -neighborhood of a set S in a metric space, i.e. $U_r(S) = \bigcup_{x \in S} \mathbb{B}_{\mathbb{X}}(x, r)$.

Definition A.6 (Hausdorff distance) Let \mathbb{X} be a metric space, and $X, Y \subset \mathbb{X}$ be a subset. The Hausdorff distance between X and Y , denoted by $d_H(X, Y)$, is defined as

$$d_H(X, Y) = \inf\{r > 0 : X \subset U_r(Y) \text{ and } Y \subset U_r(X)\}.$$

The *Gromov-Hausdorff distance* measures how two sets are far from being isometric to each other. To define the distance, we first define a relation between two sets called *correspondence*.

Definition A.7 Let X and Y be two sets. A correspondence between X and Y is a set $C \subset X \times Y$ whose projections to both X and Y are both surjective, i.e. for every $x \in X$, there exists $y \in Y$ such that $(x, y) \in C$, and for every $y \in Y$, there exists $x \in X$ with $(x, y) \in C$.

For a correspondence, we define its *distortion* by how the metric structures of two sets differ by the correspondence.

Definition A.8 Let X and Y be two metric spaces, and C be a correspondence between X and Y . The distortion of C is defined by

$$dis(C) = \sup\{|d_X(x, x') - d_Y(y, y')| : (x, y), (x', y') \in C\}.$$

Now the Gromov-Hausdorff distance [Burago et al., 2001, Theorem 7.3.25] is defined as the smallest possible distortion between two sets.

Definition A.9 (Gromov-Hausdorff distance) Let X and Y be two metric spaces. The Gromov-Hausdorff distance between X and Y , denoted as $d_{GH}(X, Y)$, is defined as

$$d_{GH}(X, Y) = \frac{1}{2} \inf_C dis(C),$$

where the infimum is over all correspondences between X and Y .

B Bottleneck distance and Wasserstein distance

Our stability bound in Theorem 4.1 is based on the bottleneck distance, while the stability bound in Hofer et al. [2017] is based on Wasserstein distance. Hence to compare these bounds, we need to understand the relationship between the bottleneck distance and Wasserstein distance. We already know that the Wasserstein distance is lower bounded by the bottleneck distance. Here, we will find a tighter lower bound for the ratio of the Wasserstein distance to the bottleneck distance.

Before analyzing the relationship between them, we first show a claim.

Claim B.1 Let $\mathcal{D}, \mathcal{D}'$ be two persistence diagrams. For $t > 0$, let $n_t \in \mathbb{N}$ be satisfying the followings: for any two diagrams $\mathcal{D}_t, \mathcal{D}'_t$ with $d_B(\mathcal{D}, \mathcal{D}_t) \leq t$ and $d_B(\mathcal{D}', \mathcal{D}'_t) \leq t$, either $|\mathcal{D}_t \setminus \mathcal{D}'_t| \geq n_t$ or $|\mathcal{D}'_t \setminus \mathcal{D}_t| \geq n_t$ holds. Then for any bijection $\gamma : \bar{\mathcal{D}} \rightarrow \bar{\mathcal{D}'}$, the number of paired points with being at least $2t$ apart in L_∞ distance is greater or equal to n_t , i.e.,

$$|\{p \in \bar{\mathcal{D}} : \|p - \gamma(p)\|_\infty > 2t\}| \geq n_t.$$

And then, we get a lower bound for the ratio of Wasserstein distance to the bottleneck distance.

Proposition B.1 Let $\mathcal{D}, \mathcal{D}'$ be two persistence diagrams. For $t > 0$, let $n_t \in \mathbb{N}$ be satisfying the followings: for any two diagrams $\mathcal{D}_t, \mathcal{D}'_t$ with $d_B(\mathcal{D}, \mathcal{D}_t) \leq t$ and $d_B(\mathcal{D}', \mathcal{D}'_t) \leq t$, either $|\mathcal{D}_t \setminus \mathcal{D}'_t| \geq n_t$ or $|\mathcal{D}'_t \setminus \mathcal{D}_t| \geq n_t$ holds. Then, the ratio of q -Wasserstein distance to the bottleneck distance is bounded as

$$\frac{W_q(\mathcal{D}, \mathcal{D}')}{d_B(\mathcal{D}, \mathcal{D}')} \geq \left(1 + \left(\frac{2t}{d_B(\mathcal{D}, \mathcal{D}')} \right)^q (n_t - 1)\right)^{\frac{1}{q}}.$$

C Stability for Vietoris-Rips and Čech filtration

When we use Vietoris-Rips or Čech filtration, our result can be turned into the stability result with respect to points in Euclidean space. Let $\mathbb{X}, \mathbb{Y} \subset \mathbb{R}^d$ be two bounded sets. The next corollary re-states our stability theorem with respect to points in \mathbb{R}^d .

Corollary C.1 Let X, Y be any ϵ -coverings of \mathbb{X}, \mathbb{Y} , and let $\mathcal{D}_X, \mathcal{D}_Y$ denote persistence diagrams induced from the Vietoris-Rips or Čech filtration on X, Y respectively. Then we have

$$|S_{\theta, \omega}(\mathcal{D}_X; \nu) - S_{\theta, \omega}(\mathcal{D}_Y; \nu)| \leq 2L_g (d_{GH}(\mathbb{X}, \mathbb{Y}) + 2\epsilon). \quad (9)$$

The proof is given in Appendix E.7. Corollary C.1 implies that if we assume our observed data points are sufficiently decent quality in the sense that $\epsilon \rightarrow 0$, then our topological layers constructed on those observed points are stable with respect to small perturbations of the true representation under proper persistent homologies. Here, ϵ could be interpreted as uncertainty from incomplete sampling. This means the topological information embedded in the proposed layer is robust against small sampling noise or data corruption by missingness.

Moreover, since Gromov-Hausdorff distance is upper bounded by Hausdorff distance, the result in Corollary C.1 also holds when we use $d_H(X, Y)$ in place of $d_{GH}(X, Y)$ in RHS of (9).

Remark 2 In fact, when we have very dense data that have been well-sampled uniformly over the true representation so that $\epsilon \rightarrow 0$, our result in (9) converges to the following:

$$|S_{\theta, \omega}(\mathcal{D}_{\mathbb{X}}; \nu) - S_{\theta, \omega}(\mathcal{D}_{\mathbb{Y}}; \nu)| \leq 2L_g d_{GH}(\mathbb{X}, \mathbb{Y}).$$

D Differentiability of DTM function

Here we provide a specific example of computing $\frac{\partial f(\varsigma)}{\partial X_j}$ when f is the DTM filtration which has not been explored in previous approaches. We first consider the case of (4) where X_j 's are data points, as in Proposition D.1. See Appendix E.8 for the proof.

Proposition D.1 When X_j 's and ς satisfy that $\sum_{X_i \in N_k(y)} \varpi_i \|X_i - y_l\|^r$ are different for each $y_l \in \varsigma$, then $f(\varsigma)$ is differentiable with respect to X_j and

$$\frac{\partial f(\varsigma)}{\partial X_j} = \frac{\varpi'_j \|X_j - y\|^{r-2} (X_j - y) I(X_j \in N_k(y))}{\left(\hat{d}_{m_0}(y)\right)^{r-1} m_0 \sum_{i=1}^n \varpi_i},$$

where I is an indicator function and $y = \arg \max_{z \in \varsigma} \hat{d}_{m_0}(z)$. In particular, f is differentiable a.e. with respect to Lebesgue measure on X .

Similarly, we consider the case of (5) where X_j 's are weights, as in Proposition D.2. See Appendix E.9 for the proof.

Proposition D.2 When X_j 's and ς satisfy that $\sum_{Y_i \in N_k(y)} X'_i \|Y_i - y_l\|^r$ are different for each $y_l \in \varsigma$, then $f(\varsigma)$ is differentiable with respect to X_j and

$$\frac{\partial f(\varsigma)}{\partial X_j} = \frac{\|Y_j - y\|^r I(Y_j \in N_k(y)) - m_0 \left(\hat{d}_{m_0}(y) \right)^r}{r \left(\hat{d}_{m_0}(y) \right)^{r-1} m_0 \sum_{i=1}^n X_i},$$

where $y = \arg \max_{y \in \varsigma_i} \hat{d}_{m_0}(y)$. In particular, f is differentiable a.e. with respect to Lebesgue measure on X and Y .

Computation of $\frac{\partial h_{\text{top}}}{\partial \mu_i}$, $\frac{\partial h_{\text{top}}}{\partial \varsigma_i}$ are simpler and can be done in a similar fashion. In the experiments, we set $r = 2$.

E Proofs

E.1 Proof of Theorem 3.1

For computing $\frac{\partial h_{\text{top}}}{\partial X_j}$, note that it can be expanded using the chain rule as

$$\frac{\partial h_{\text{top}}}{\partial X_j} = \sum_i \frac{\partial h_{\text{top}}}{\partial b_i} \frac{\partial b_i}{\partial X_j} + \sum_i \frac{\partial h_{\text{top}}}{\partial d_i} \frac{\partial d_i}{\partial X_j}, \quad (10)$$

and hence we need to compute $\frac{\partial \mathcal{D}_X}{\partial X} = \left\{ \left(\frac{\partial b_i}{\partial X_j}, \frac{\partial d_i}{\partial X_j} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X, X_j \in X}$ and $\frac{\partial h_{\text{top}}}{\partial \mathcal{D}_X} = \left\{ \left(\frac{\partial h_{\text{top}}}{\partial b_i}, \frac{\partial h_{\text{top}}}{\partial d_i} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X}$ to compute $\frac{\partial h_{\text{top}}}{\partial X_j}$.

We first compute $\frac{\partial \mathcal{D}_X}{\partial X}$. Let K be the simplicial complex, and suppose all the simplices are ordered in the filtration so that the values of f are nondecreasing, i.e. if ς comes earlier than τ then $f(\varsigma) \leq f(\tau)$. Note that the map ξ from each birth-death point $(b_i, d_i) \in \mathcal{D}_X$ to a pair of simplices (β_i, δ_i) is simply the pairing returned by the standard persistence diagram [Carlsson et al., 2005]. Let γ be the homological feature corresponding to (b_i, d_i) , then the birth simplex β_i is the simplex that forms γ in $K_{b_i} = f^{-1}(-\infty, b_i]$, and the death simplex δ_i is the simplex that causes γ to collapse in $K_{d_i} = f^{-1}(-\infty, d_i]$. For example, if γ were to be a 1-dimensional feature, then β_i is the edge in K_{b_i} that forms the loop corresponding to γ , and δ_i is the triangle in K_{d_i} which incurs the loop corresponding to γ can be contracted in K_{d_i} .

Now, $f(\xi(b_i)) = f(\beta_i) = b_i$ and $f(\xi(d_i)) = f(\delta_i) = d_i$, and from ξ being locally constant on X ,

$$\frac{\partial b_i}{\partial X_j} = \frac{\partial f(\xi(b_i))}{\partial X_j} = \frac{\partial f(\beta_i)}{\partial X_j}, \quad \frac{\partial d_i}{\partial X_j} = \frac{\partial f(\xi(d_i))}{\partial X_j} = \frac{\partial f(\delta_i)}{\partial X_j}. \quad (11)$$

Therefore, the derivatives of the birth value and the death value are the derivatives of the filtration function evaluated at the corresponding pair of simplices. And $\frac{\partial \mathcal{D}_X}{\partial X} = \left\{ \left(\frac{\partial b_i}{\partial X_j}, \frac{\partial d_i}{\partial X_j} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X, X_j \in X}$ is the collection of these derivatives, hence applying (11) gives

$$\frac{\partial \mathcal{D}_X}{\partial X} = \left\{ \left(\frac{\partial b_i}{\partial X_j}, \frac{\partial d_i}{\partial X_j} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X, X_j \in X} = \left\{ \left(\frac{\partial f(\beta_i)}{\partial X_j}, \frac{\partial f(\delta_i)}{\partial X_j} \right) \right\}_{\xi^{-1}(\beta_i, \delta_i) \in \mathcal{D}_X, X_j \in X}. \quad (12)$$

Now, we compute $\frac{\partial h_{\text{top}}}{\partial \mathcal{D}_X} = \left\{ \left(\frac{\partial h_{\text{top}}}{\partial b_i}, \frac{\partial h_{\text{top}}}{\partial d_i} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X}$. Computing $\frac{\partial h_{\text{top}}}{\partial b_i}$ can be done by applying the chain rule on $h_{\text{top}} = S_{\theta, \omega} = g_{\theta} \circ \bar{\Lambda}_{\omega}$ as

$$\frac{\partial h_{\text{top}}}{\partial b_i} = \frac{\partial S_{\theta, \omega}}{\partial b_i} = \frac{\partial (g_{\theta} \circ \bar{\Lambda}_{\omega})}{\partial b_i} = \nabla g_{\theta} \circ \frac{\partial \bar{\Lambda}_{\omega}}{\partial b_i} = \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \frac{\partial \bar{\Lambda}_{\omega}(lv)}{\partial b_i}, \quad (13)$$

where we use x_l as the shorthand notation for the input of the function g_{θ} . Then, applying $\bar{\Lambda}_{\omega}(lv) = \sum_{k=1}^{K_{\max}} \omega_k \lambda_k(lv)$ to (13) gives

$$\frac{\partial h_{\text{top}}}{\partial b_i} = \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial b_i}. \quad (14)$$

Similarly, $\frac{\partial h_{\text{top}}}{\partial d_i}$ can be computed as

$$\frac{\partial h_{\text{top}}}{\partial d_i} = \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial d_i}. \quad (15)$$

And therefore, $\frac{\partial h_{\text{top}}}{\partial \mathcal{D}_X}$ is the collection of these derivatives from (14) and (15), i.e.,

$$\frac{\partial h_{\text{top}}}{\partial \mathcal{D}_X} = \left\{ \left(\sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial b_i}, \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial d_i} \right) \right\}_{(b_i, d_i) \in \mathcal{D}_X}. \quad (16)$$

Hence, $\frac{\partial h_{\text{top}}}{\partial X}$ can be computed by applying (12) and (16) to (10) as

$$\begin{aligned} \frac{\partial h_{\text{top}}}{\partial X_j} &= \sum_i \frac{\partial h_{\text{top}}}{\partial b_i} \frac{\partial b_i}{\partial X_j} + \sum_i \frac{\partial h_{\text{top}}}{\partial d_i} \frac{\partial d_i}{\partial X_j} \\ &= \sum_i \frac{\partial f(\beta_i)}{\partial X_j} \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial b_i} + \sum_i \frac{\partial f(\delta_i)}{\partial X_j} \sum_{l=1}^m \frac{\partial g_{\theta}}{\partial x_l} \sum_{k=1}^{K_{\max}} \omega_k \frac{\partial \lambda_k(lv)}{\partial d_i}. \end{aligned}$$

E.2 Proof of Theorem 4.1

Let \mathcal{D} and \mathcal{D}' be two persistence diagrams and let λ and λ' be their persistence landscapes. All the quantities derived from \mathcal{D}' are denoted by a variable name with the superscript $'$ hereafter (e.g., $\lambda'_k(t), \bar{\Lambda}'_{\omega}$).

For the stability of the structure element $S_{\theta, \omega}$, we first expand the difference between $S_{\theta, \omega}(\mathcal{D}; \nu)$ and $S_{\theta, \omega}(\mathcal{D}'; \nu)$ using $S_{\theta, \omega} = g_{\theta} \circ \bar{\Lambda}_{\omega}$ as

$$|S_{\theta, \omega}(\mathcal{D}; \nu) - S_{\theta, \omega}(\mathcal{D}'; \nu)| = |g_{\theta}(\bar{\Lambda}_{\omega}) - g_{\theta}(\bar{\Lambda}'_{\omega})|. \quad (17)$$

Then, RHS of (17) is bounded by applying the Lipschitz condition of the function g_{θ} as

$$|g_{\theta}(\bar{\Lambda}_{\omega}) - g_{\theta}(\bar{\Lambda}'_{\omega})| \leq L_g \|\bar{\Lambda}_{\omega} - \bar{\Lambda}'_{\omega}\|_{\infty}. \quad (18)$$

Then for $\|\bar{\Lambda}_{\omega} - \bar{\Lambda}'_{\omega}\|_{\infty}$, note that $\bar{\Lambda}_{\omega}, \bar{\Lambda}'_{\omega} \in \mathbb{R}^m$, the L_{∞} difference of $\bar{\Lambda}_{\omega}$ and $\bar{\Lambda}'_{\omega}$ is bounded as

$$\begin{aligned} \|\bar{\Lambda}_{\omega} - \bar{\Lambda}'_{\omega}\|_{\infty} &= \max_{0 \leq i \leq m-1} |\bar{\lambda}_{\omega}(T_{\min} + i\nu) - \bar{\lambda}'_{\omega}(T_{\min} + i\nu)| \\ &\leq \sup_{t \in [0, T]} |\bar{\lambda}_{\omega}(t) - \bar{\lambda}'_{\omega}(t)| = m^{1/2} \|\bar{\lambda}_{\omega} - \bar{\lambda}'_{\omega}\|_{\infty}. \end{aligned} \quad (19)$$

Now, for bounding $\|\bar{\lambda}_{\omega} - \bar{\lambda}'_{\omega}\|_{\infty}$, we first consider the pointwise difference $|\bar{\lambda}_{\omega}(t) - \bar{\lambda}'_{\omega}(t)|$. For all $t \in [0, T]$, the difference between $\bar{\lambda}_{\omega}(t)$ and $\bar{\lambda}'_{\omega}(t)$ is bounded as

$$\begin{aligned} |\bar{\lambda}_{\omega}(t) - \bar{\lambda}'_{\omega}(t)| &= \left| \frac{1}{\sum_k \omega_k} \sum_{k=1}^{K_{\max}} \omega_k \lambda_k(t) - \frac{1}{\sum_k \omega_k} \sum_{k=1}^{K_{\max}} \omega_k \lambda'_k(t) \right| \\ &\leq \frac{1}{\sum_k \omega_k} \sum_{k=1}^{K_{\max}} \omega_k |\lambda_k(t) - \lambda'_k(t)| \\ &\leq \sup_{1 \leq k \leq K_{\max}, t \in [0, T]} |\lambda_k(t) - \lambda'_k(t)| = \max_{1 \leq k \leq K_{\max}} \|\lambda_k - \lambda'_k\|_{\infty}. \end{aligned} \quad (20)$$

And hence $\|\bar{\lambda}_{\omega} - \bar{\lambda}'_{\omega}\|_{\infty}$ is bounded by $\max_{1 \leq k \leq K_{\max}} \|\lambda_k - \lambda'_k\|_{\infty}$ as well, i.e.,

$$\|\bar{\lambda}_{\omega} - \bar{\lambda}'_{\omega}\|_{\infty} = \sup_{t \in [0, T]} |\bar{\lambda}_{\omega}(t) - \bar{\lambda}'_{\omega}(t)| \leq \max_{1 \leq k \leq K_{\max}} \|\lambda_k - \lambda'_k\|_{\infty}. \quad (21)$$

Then for all $k = 1, \dots, K_{\max}$, the ∞ -landscape distance $\|\lambda_k - \lambda'_k\|_\infty$ is bounded by the bottleneck distance $d_B(\mathcal{D}, \mathcal{D}')$ from Theorem 13 in Bubenik [2015], i.e.

$$\|\lambda_k - \lambda'_k\|_\infty \leq d_B(\mathcal{D}, \mathcal{D}'). \quad (22)$$

Hence, applying (18), (19), (21), (22) to (17) gives the stated stability result as

$$\begin{aligned} |S_{\theta, \omega}(\mathcal{D}; \nu) - S_{\theta, \omega}(\mathcal{D}'; \nu)| &= \left| g_\theta(\overline{\Lambda}_\omega) - g_\theta(\overline{\Lambda}'_\omega) \right| \leq L_g \left\| \overline{\Lambda}_\omega - \overline{\Lambda}'_\omega \right\|_\infty \\ &\leq L_g \left\| \overline{\lambda}_\omega - \overline{\lambda}'_\omega \right\|_\infty \leq L_g \max_{1 \leq k \leq K_{\max}} \|\lambda_k - \lambda'_k\|_\infty \\ &\leq L_g d_B(\mathcal{D}, \mathcal{D}'). \end{aligned}$$

E.3 Proof of Corollary 4.1

First note that the result of Hofer et al. [2017] used W_1 Wasserstein distance with L_r norm for $\forall r \in \mathbb{N}$, which will be denoted by $W_1^{L_r}$ in this proof. That is,

$$W_1^{L_r}(\mathcal{D}, \mathcal{D}') := \inf_{\gamma} \sum_{p \in \mathcal{D}_X} \|p - \gamma(p)\|_r$$

where γ ranges over all bijections $\mathcal{D} \rightarrow \mathcal{D}'$ (i.e., $W_1^{L_\infty}$ corresponds to W_1 in our definition 2.2). Then, $\|\cdot\|_r \geq \|\cdot\|_\infty$ implies that $W_1^{L_r}$ is lower bounded by W_1 , i.e.

$$W_1^{L_r}(\mathcal{D}, \mathcal{D}') \geq W_1(\mathcal{D}, \mathcal{D}'). \quad (23)$$

Now, let c_K denote the Lipschitz constant in Hofer et al. [2017, Theorem 1] and c_{g_θ} denote the constant term in our result in Theorem 4.1, i.e. $c_{g_\theta} = L_g \left(\frac{T}{\nu}\right)^{1/2}$. We want to upper bound the ratio $\frac{c_{g_\theta} d_B(\mathcal{D}, \mathcal{D}')}{c_K W_1^{L_r}(\mathcal{D}, \mathcal{D}')}$. This directly comes from (23) and Proposition B.1 as

$$\frac{c_{g_\theta} d_B(\mathcal{D}, \mathcal{D}')}{c_K W_1^{L_r}(\mathcal{D}, \mathcal{D}')} \geq \frac{c_{g_\theta}}{c_K} \frac{d_B(\mathcal{D}, \mathcal{D}')}{W_1(\mathcal{D}, \mathcal{D}')} \geq \frac{c_{g_\theta}}{c_K} \frac{1}{1 + \frac{2t}{d_B(\mathcal{D}, \mathcal{D}')} (n_t - 1)}.$$

Finally, we define $C_{g_\theta, T, \nu} := \frac{c_{g_\theta, T, \nu}}{c_K}$, and the result follows.

It should be noted that the bound is actually very loose. However, we can still conclude that our bound is tighter than that of Hofer et al. [2017] at polynomial rates.

E.4 Proof of Theorem 4.2

We first bound the difference between $S_{\theta, \omega}(\mathcal{D}_X; \nu)$ and $S_{\theta, \omega}(\mathcal{D}_P; \nu)$ using Theorem 4.1 as

$$|S_{\theta, \omega}(\mathcal{D}_X; \nu) - S_{\theta, \omega}(\mathcal{D}_P; \nu)| \leq L_g d_B(\mathcal{D}_X, \mathcal{D}_P). \quad (24)$$

It is left to further bound the bottleneck distance $d_B(\mathcal{D}_X, \mathcal{D}_P)$. The bottleneck distance between two diagrams \mathcal{D}_X and \mathcal{D}_P is bounded by the stability theorem of persistent homology as

$$d_B(\mathcal{D}_X, \mathcal{D}_P) \leq \|d_{P_n, m_0} - d_{P, m_0}\|_\infty. \quad (25)$$

Then, from $r = 2$ in the DTM function, the L_∞ distance between d_{P_n, m_0} and d_{P, m_0} is bounded by the stability of DTM function (Theorem 3.5 from Chazal et al. [2011]) as

$$\|d_{P_n, m_0} - d_{P, m_0}\|_\infty \leq m_0^{-1/2} W_2(P_n, P). \quad (26)$$

Hence, combining (24), (25), and (26) altogether gives the stated stability result as

$$|S_{\theta, \omega}(\mathcal{D}_X; \nu) - S_{\theta, \omega}(\mathcal{D}_P; \nu)| \leq L_g m_0^{-1/2} W_2(P_n, P).$$

E.5 Proof of Claim B.1

Let $\gamma: \mathcal{D} \rightarrow \mathcal{D}'$ be any bijection and let $\mathcal{S} := \{p \in \bar{\mathcal{D}} : \|p - \gamma(p)\|_\infty > 2t\}$. Then for $p \in \bar{\mathcal{D}}$ with $\|p - \gamma(p)\|_\infty \leq 2t$, there exists $\beta(p) \in \mathbb{R}_*^2$ such that $\|p - \beta(p)\|_\infty \leq t$ and $\|\beta(p) - \gamma(p)\|_\infty \leq t$. Now, define two diagrams $\mathcal{D}_t, \mathcal{D}'_t$ as follows:

$$\begin{aligned}\mathcal{D}_t &= \mathcal{S} \cup \{\beta(p) : p \in \bar{\mathcal{D}} \setminus \mathcal{S}\} \setminus \text{Diag}, \\ \mathcal{D}'_t &= \mathcal{S}' \cup \{\beta(p) : p \in \bar{\mathcal{D}} \setminus \mathcal{S}\} \setminus \text{Diag},\end{aligned}$$

where $\mathcal{S}' := \{\gamma(p) : p \in \bar{\mathcal{D}}\}$. Then, $d_B(\mathcal{D}, \mathcal{D}_t) \leq t$ and $d_B(\mathcal{D}', \mathcal{D}'_t) \leq t$ from the construction. Hence from the definition of n_t , either $|\mathcal{D}_t \setminus \mathcal{D}'_t| \geq n_t$ or $|\mathcal{D}'_t \setminus \mathcal{D}_t| \geq n_t$ holds. Now, note that

$$\mathcal{D}_t \setminus \mathcal{D}'_t \subset \mathcal{S} \quad \text{and} \quad \mathcal{D}'_t \setminus \mathcal{D}_t \subset \mathcal{S}'.$$

And $|\mathcal{S}| = |\mathcal{S}'|$, and hence we get the claimed result as

$$|\mathcal{S}| \geq n_t.$$

E.6 Proof of Proposition B.1

We consider a bijection γ^* that realizes the q -Wasserstein distance between \mathcal{D} and \mathcal{D}' : i.e. $\gamma^* = \underset{\gamma}{\operatorname{arginf}} \sum_{p \in \mathcal{D}} \|p - \gamma(p)\|_\infty^q$. Then we have that

$$d_B(\mathcal{D}, \mathcal{D}')^q \leq \sup_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q. \quad (27)$$

On the other hand, if we let $p^* = \operatorname{argsup}_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty$, we have

$$W_q(\mathcal{D}, \mathcal{D}')^q = \sum_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q = \sup_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q + \sum_{p \neq p^*} \|p - \gamma^*(p)\|_\infty^q.$$

Note that from Claim B.1, $|\{p \in \bar{\mathcal{D}} : \|p - \gamma^*(p)\|_\infty > 2t\}| \geq n_t$. And hence $W_q(\mathcal{D}, \mathcal{D}')^q$ can be lower bounded as

$$W_q(\mathcal{D}, \mathcal{D}')^q = \sup_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q + \sum_{p \neq p^*} \|p - \gamma^*(p)\|_\infty^q \quad (28)$$

$$\geq \sup_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q + (2t)^q(n_t - 1). \quad (29)$$

Now, we lower bound the ratio $\frac{W_q(\mathcal{D}, \mathcal{D}')^q}{d_B(\mathcal{D}, \mathcal{D}')^q}$. By (27) and (29), this can be done as follows.

$$\begin{aligned}\frac{W_q(\mathcal{D}, \mathcal{D}')^q}{d_B(\mathcal{D}, \mathcal{D}')^q} &\geq \frac{\sup_{p \in \mathcal{D}} \|p - \gamma^*(p)\|_\infty^q + (2t)^q(n_t - 1)}{d_B(\mathcal{D}, \mathcal{D}')^q} \\ &\geq 1 + \left(\frac{2t}{d_B(\mathcal{D}, \mathcal{D}')} \right)^q (n_t - 1).\end{aligned}$$

And hence the ratio of the Wasserstein distance to the bottleneck distance $\frac{W_q(\mathcal{D}, \mathcal{D}')}{d_B(\mathcal{D}, \mathcal{D}')}$ is correspondingly lower bounded as

$$\frac{W_q(\mathcal{D}, \mathcal{D}')}{d_B(\mathcal{D}, \mathcal{D}')} \geq \left(\frac{W_q(\mathcal{D}, \mathcal{D}')^q}{d_B(\mathcal{D}, \mathcal{D}')^q} \right)^{\frac{1}{q}} \geq \left(1 + \left(\frac{2t}{d_B(\mathcal{D}, \mathcal{D}')} \right)^q (n_t - 1) \right)^{\frac{1}{q}}.$$

E.7 Proof of Corollary C.1

The difference between $S_{\theta, \omega}(\mathcal{D}_X; \nu)$ and $S_{\theta, \omega}(\mathcal{D}_Y; \nu)$ is bounded by Theorem 4.1 as

$$|S_{\theta, \omega}(\mathcal{D}_X; \nu) - S_{\theta, \omega}(\mathcal{D}_Y; \nu)| \leq L_g d_B(\mathcal{D}_X, \mathcal{D}_Y), \quad (30)$$

hence it suffices to show

$$d_B(\mathcal{D}_X, \mathcal{D}_Y) < 2(d_{GH}(\mathbb{X}, \mathbb{Y}) + 2\epsilon). \quad (31)$$

To show (31), we first apply the triangle inequality as

$$d_B(\mathcal{D}_X, \mathcal{D}_Y) \leq d_B(\mathcal{D}_X, \mathcal{D}_{\mathbb{X}}) + d_B(\mathcal{D}_{\mathbb{X}}, \mathcal{D}_{\mathbb{Y}}) + d_B(\mathcal{D}_{\mathbb{Y}}, \mathcal{D}_Y). \quad (32)$$

And note that since $\mathbb{X}, \mathbb{Y}, X, Y$ are all bounded in Euclidean space, they are totally bounded metric spaces. Thus by Theorem 5.2 in Chazal et al. [2014a], the bottleneck distance between any two diagrams is bounded by Gromov-Hausdorff distance, and in particular,

$$\begin{aligned} d_B(\mathcal{D}_{\mathbb{X}}, \mathcal{D}_{\mathbb{Y}}) &\leq 2d_{GH}(\mathbb{X}, \mathbb{Y}), \\ d_B(\mathcal{D}_X, \mathcal{D}_{\mathbb{X}}) &\leq 2d_{GH}(X, \mathbb{X}), \quad d_B(\mathcal{D}_{\mathbb{Y}}, \mathcal{D}_Y) \leq 2d_{GH}(\mathbb{Y}, Y). \end{aligned} \quad (33)$$

And then since the Gromov-Hausdorff distance is bounded by the Hausdorff distance,

$$d_{GH}(X, \mathbb{X}) \leq d_H(X, \mathbb{X}), \quad d_{GH}(\mathbb{Y}, Y) \leq d_H(\mathbb{Y}, Y). \quad (34)$$

And the Hausdorff distance between X and \mathbb{X} or Y and \mathbb{Y} is bounded by ϵ by the assumption that X, Y are ϵ -coverings of \mathbb{X}, \mathbb{Y} , respectively, i.e.,

$$d_H(X, \mathbb{X}) < \epsilon, \quad d_H(\mathbb{Y}, Y) < \epsilon. \quad (35)$$

Hence combining (32), (33), (34), and (35) gives (31) as

$$\begin{aligned} d_B(\mathcal{D}_X, \mathcal{D}_Y) &\leq d_B(\mathcal{D}_X, \mathcal{D}_{\mathbb{X}}) + d_B(\mathcal{D}_{\mathbb{X}}, \mathcal{D}_{\mathbb{Y}}) + d_B(\mathcal{D}_{\mathbb{Y}}, \mathcal{D}_Y) \\ &\leq 2(d_{GH}(X, \mathbb{X}) + d_{GH}(\mathbb{X}, \mathbb{Y}) + d_{GH}(\mathbb{Y}, Y)) \\ &\leq 2(d_H(X, \mathbb{X}) + d_{GH}(\mathbb{X}, \mathbb{Y}) + d_H(\mathbb{Y}, Y)) \\ &< 2(d_{GH}(\mathbb{X}, \mathbb{Y}) + 2\epsilon). \end{aligned}$$

Now, the results follows from (30) and (31).

E.8 Proof of Proposition D.1

From (4), note that for any $y \in \varsigma$, $\hat{d}_{m_0}(y)$ is expanded as

$$\hat{d}_{m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} \varpi'_i \|X_i - y\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{1/r}, \quad (36)$$

where k is such that $\sum_{X_i \in N_{k-1}(y)} \varpi_i < m_0 \sum_{i=1}^n \varpi_i \leq \sum_{X_i \in N_k(y)} \varpi_i$, and $\varpi'_i = \sum_{X_j \in N_k(y)} \varpi_j - m_0 \sum_{j=1}^n \varpi_j$ for one of X_i 's that is k -th nearest neighbor of y and $\varpi'_i = \varpi_i$ otherwise. Hence, by letting $y = \arg \max_{z \in \varsigma} \hat{d}_{m_0}(z)$ applying to (36), the filtration function f_X at simplex ς becomes

$$f_X(\varsigma) = \hat{d}_{X, m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} \varpi'_i \|X_i - y\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{1/r}, \quad (37)$$

where the notations f_X and \hat{d}_{X, m_0} are to clarify the dependency of f on X . And from the condition, $\hat{d}_{m_0}(y) > \hat{d}_{m_0}(z)$ holds for all $z \in \varsigma$. Hence for sufficiently small $\epsilon > 0$ and for any $Z' = \{Z_1, \dots, Z_n\}$ with $\|Z_j - X_j\| < \epsilon$, (37) becomes

$$f_Z(\varsigma) = \hat{d}_{Z, m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} \varpi'_i \|Z_i - y\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{1/r}. \quad (38)$$

Hence by differentiating (38), the derivative of f with respect to X is calculated as

$$\begin{aligned} \frac{\partial f(\varsigma)}{\partial X_j} &= \left(\frac{\sum_{X_i \in N_k(y)} \varpi'_i \|X_i - y\|^r}{m_0 \sum_{i=1}^n \varpi_i} \right)^{\frac{1}{r}-1} \times \frac{\varpi'_j \|X_j - y\|^{r-2} (X_j - y) I(X_j \in N_k(y))}{m_0 \sum_{i=1}^n \varpi_i} \\ &= \frac{\varpi'_j \|X_j - y\|^{r-2} (X_j - y) I(X_j \in N_k(y))}{\left(\hat{d}_{m_0}(y) \right)^{r-1} m_0 \sum_{i=1}^n \varpi_i}. \end{aligned}$$

E.9 Proof of Proposition D.2

From (5), note that for any $y \in \varsigma$, $\hat{d}_{m_0}(y)$ is expanded as

$$\hat{d}_{m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} X'_i \|Y_i - y\|^r}{m_0 \sum_{i=1}^n X_i} \right)^{1/r}, \quad (39)$$

where k is such that $\sum_{Y_i \in N_{k-1}(y)} X_i < m_0 \sum_{i=1}^n X_i \leq \sum_{Y_i \in N_k(y)} X_i$, and $X'_i = \sum_{X_j \in N_k(y)} X_j - m_0 \sum_{j=1}^n X_j$ for one of Y_i 's that is k -th nearest neighbor of y and $X'_i = X_i$ otherwise. Hence, by letting $y = \arg \max_{z \in \varsigma} \hat{d}_{m_0}(z)$ and applying to (39), the filtration function f_X at simplex ς becomes

$$f_X(\varsigma) = \hat{d}_{X, m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} X'_i \|Y_i - y\|^r}{m_0 \sum_{i=1}^n X_i} \right)^{1/r}, \quad (40)$$

where the notations f_X and \hat{d}_{X, m_0} are to clarify the dependency of f on X . And from the condition, $\hat{d}_{m_0}(y) > \hat{d}_{m_0}(z)$ holds for all $z \in \varsigma$. Hence for sufficiently small $\epsilon > 0$ and for any $Z' = \{Z_1, \dots, Z_n\}$ with $\|Z_j - X_j\| < \epsilon$, (40) becomes

$$f_Z(\varsigma) = \hat{d}_{Z, m_0}(y) = \left(\frac{\sum_{X_i \in N_k(y)} Z'_i \|Y_i - y\|^r}{m_0 \sum_{i=1}^n Z_i} \right)^{1/r}. \quad (41)$$

Hence by differentiating (41), the derivative of f with respect to X is calculated as

$$\begin{aligned} & \frac{\partial f(\varsigma)}{\partial X_j} \\ &= \frac{1}{r} \left(\frac{\sum_{X_i \in N_k(y)} X'_i \|Y_i - y\|^r}{m_0 \sum_{i=1}^n X_i} \right)^{\frac{1}{r}-1} \times \\ & \quad \frac{\|Y_j - y\|^r I(Y_j \in N_k(y)) (m_0 \sum_{i=1}^n X_i) - m_0 \left(\sum_{X_i \in N_k(y)} X'_i \|Y_i - y\|^r \right)}{(m_0 \sum_{i=1}^n X_i)^2} \\ &= \frac{\|Y_j - y\|^r I(Y_j \in N_k(y)) - m_0 \left(\hat{d}_{m_0}(y) \right)^r}{r \left(\hat{d}_{m_0}(y) \right)^{r-1} m_0 \sum_{i=1}^n X_i}. \end{aligned}$$

F Guideline for choosing TDA parameters

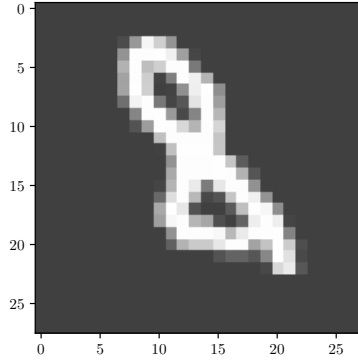
PLay has several TDA parameters to choose: K_{\max} , T_{\min} , T_{\max} , m , and m_0 if DTM filtration is used. One can try grid search but it could be too time-consuming. More affordable approach is to compute the DTM filtration and the persistence diagram for some data and choose appropriate parameters that can reveal the topological and geometrical information of the data. Figure 5 illustrates one example of the digit 8 in MNIST data. Figure 5(a) shows the contour plot of the chosen data.

When using a DTM filtration, we need to choose m_0 first. DTMs with different m_0 values extract different topological and geometrical information. When m_0 is small, a DTM filtration aggregates the data more locally, and the geometrical and homological information formed from the local structure is extracted. When m_0 is large, a DTM filtration aggregates the data more globally, and the geometrical and homological information formed from the global structure is extracted. From the digit 8, we would first like to see the two-loop structure. And if we choose $m_0 = 0.05$, then as can be seen in Figure 5(b) and (c), the 1st persistent homology extracts the two-loop structure, which is more directly expected from the contour plot of the data itself in Figure 5(a). However, if we choose $m_0 = 0.2$, then as can be seen in Figure 5(d) and (e), the two-loop structure disappears, since the two-loop structure is coming from more local geometry of the data. Meanwhile, as the DTM filtration aggregates the data more globally, the global geometry information that three points on the digit 8 (top, center, bottom) being close to neighboring points and being centers of local clusters is extracted in the 0th persistent homology. For MNIST data, DTM filtrations with $m_0 = 0.05$ and $m_0 = 0.2$ extract different topological and geometrical information of the data. Hence for MNIST data, we used two parallel PLLays with $m_0 = 0.05$ and $m_0 = 0.2$, respectively.

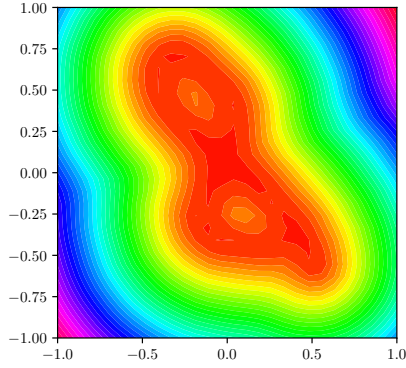
After choosing m_0 , choosing other TDA parameters K_{\max} , T_{\min} , T_{\max} , m is more straightforward. One can choose parameters so that the desired topological features are well extracted in the landscape. For $m_0 = 0.05$, as can be seen from Figure 5(c), choosing $K_{\max} = 2$, $T_{\min} = 0.06$, $T_{\max} = 0.3$, $m = 25$ will extract two 1-dimensional features of the persistence diagram in the corresponding landscape. For $m_0 = 0.2$, as can be seen from Figure 5(e), choosing $K_{\max} = 3$, $T_{\min} = 0.14$, $T_{\max} = 0.4$, $m = 27$ will extract two 1-dimensional features of the persistence diagram in the corresponding landscape.

G Experiment Details.

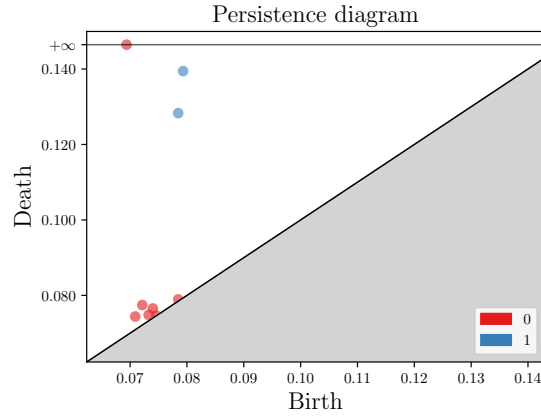
All the experiments were implemented using GUDHI The GUDHI Project [2020] and Tensorflow library in Python and TDA package Fasy et al. [2014] in R. We use mean and standard deviation across 20 runs of simulations with different network initializations. We remark that the basic purpose of our experiment design is to highlight the prospects and possibilities of using topological layer, not to win state-of-the-art performances.



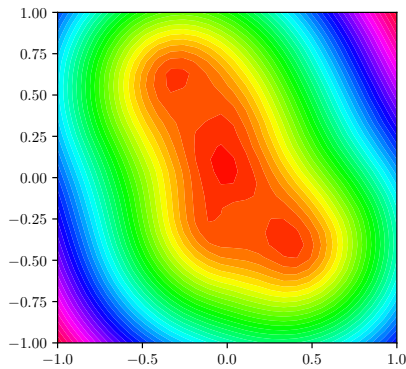
(a) Digit 8 in MNIST data.



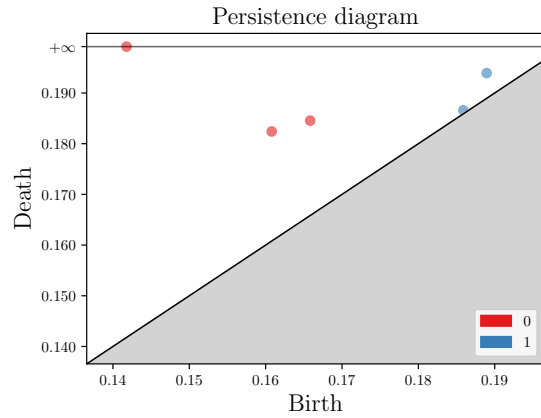
(b) Contour plot of DTM filtration, $m_0 = 0.05$.



(c) Persistence Diagram of DTM filtration, $m_0 = 0.05$.



(d) Contour plot of DTM filtration, $m_0 = 0.2$.



(e) Persistence Diagram of DTM filtration, $m_0 = 0.2$.

Figure 5: One example of the digit 8 in MNIST data, its contour plots and persistence diagrams of DTM filtration at $m_0 = 0.05$ and $m_0 = 0.2$. When $m_0 = 0.05$, DTM filtration aggregates more locally, and the 1st persistent homology extracts two loop structures of the digit 8. When $m_0 = 0.2$, DTM filtration aggregates the digit 8 more globally, and the 0th persistent homology extracts three connected component structures of the digit 8.

	Corruption and noise probability							
	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
MLP	0.8683 (0.0063)	0.8425 (0.0061)	0.8133 (0.0087)	0.7850 (0.0086)	0.7441 (0.0098)	0.6997 (0.0090)	0.6514 (0.0124)	0.5732 (0.0155)
MLP+S	0.8597 (0.0087)	0.8322 (0.0086)	0.8060 (0.0152)	0.7749 (0.0147)	0.7364 (0.0177)	0.6844 (0.0187)	0.6372 (0.0213)	0.5637 (0.0161)
MLP+P	0.8791 (0.0062)	0.8538 (0.0061)	0.8227 (0.0103)	0.7910 (0.0121)	0.7511 (0.0109)	0.7045 (0.0087)	0.6507 (0.0120)	0.5753 (0.0135)
CNN	0.8506 (0.0261)	0.8367 (0.0246)	0.8030 (0.0315)	0.7872 (0.0340)	0.7541 (0.0319)	0.7315 (0.0447)	0.6778 (0.0506)	0.6245 (0.0478)
CNN+S	0.8544 (0.0194)	0.8058 (0.1081)	0.7988 (0.0252)	0.7938 (0.0326)	0.7649 (0.0215)	0.7055 (0.1268)	0.6884 (0.0372)	0.6281 (0.0407)
CNN+P	0.8790 (0.0151)	0.8541 (0.0218)	0.8364 (0.0214)	0.8209 (0.0217)	0.7855 (0.0247)	0.7551 (0.0289)	0.7044 (0.0230)	0.6355 (0.0404)
CNN+P(i)	0.8635 (0.0189)	0.8391 (0.0153)	0.8113 (0.0250)	0.7985 (0.0275)	0.7671 (0.0179)	0.7391 (0.0302)	0.6841 (0.0936)	0.6364 (0.0355)

Table 2: Test accuracy in MNIST experiments. In each cell, the top number corresponds to the average accuracy of the model at the corruption and noise probability, and the bottom number corresponds to the 1 standard deviation of the accuracies. At each column, the model with the best accuracy is bolded.

G.1 MNIST handwritten digits.

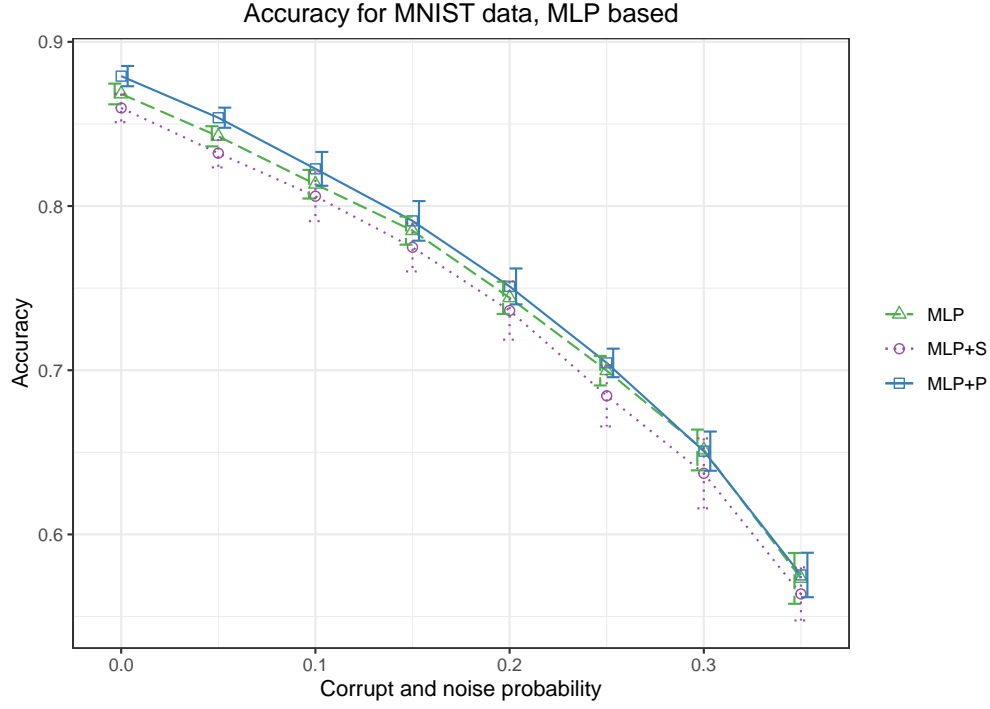
For MNIST handwritten digits, we use MNIST dataset. Raw input data is a 784 dimensional vector (reshaped from 28 by 28) of real values, each value being the pixel intensity. We use 1000 random samples for the training set and 10000 samples for the test set. Cross-entropy loss was used to train the network for 100 epochs, using Adam optimizer with mini-batches of size 16.

Topological layer. For MLP+P and CNN+P(i), we use two parallel PLLays at the beginning of MLP and CNN models with 32 nodes each and affine transformation, which are concatenated to the raw input to either MLP or CNN. We used the empirical DTM filtration in (5), where we define fixed 28×28 points on grid on $[-1, 1]^2$ and use X as a weight vector for the fixed points. For one PLLay, we used $m_0 = 0.05$, $K_{\max} = 2$, $T_{\min} = 0.06$, $T_{\max} = 0.3$, $m = 25$, and for the other PLLay, we used $m_0 = 0.2$, $K_{\max} = 3$, $T_{\min} = 0.14$, $T_{\max} = 0.4$, $m = 27$. For CNN+P, we additionally use one PLLay after the convolutional layer, with $K_{\max} = 3$, $T_{\min} = 0.05$, $T_{\max} = 0.95$, $m = 18$.

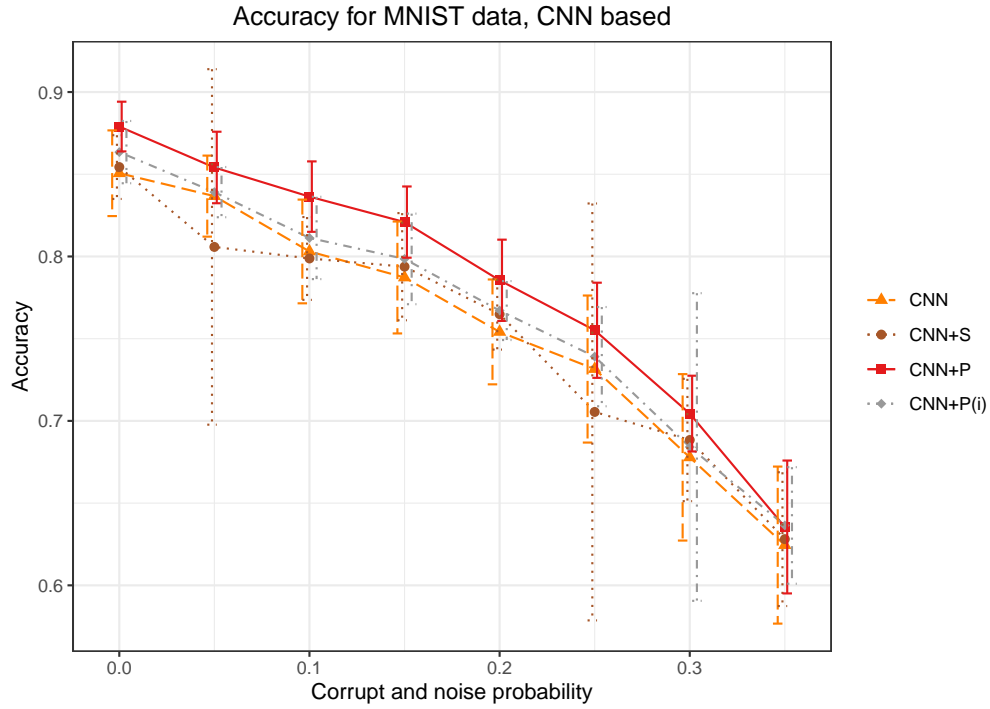
Baselines. For the baselines, models were designed to have simple structures for quick comparisons:

- Vanilla MLP: one hidden layer with 64 units with ReLU activations.
- CNN: two convolution layers followed by two fully connected layers.
- SLay: for comparison with PLLay, two SLays are used with 10 nodes each, which are concatenated to the raw input to either MLP or CNN. We used the value $\nu = 0.005$ and $\nu = 0.01$ for the hyperparameter of each SLay, respectively.

Result. The Accuracy results for MNIST data in Figure 4 is represented with 1 standard errors in Table 2 and Figure 6. In Figure 6, the results for MLP, MLP+S, MLP+P are in Figure 6(a), and the results for CNN, CNN+S, CNN+P, CNN+P(i) are in Figure 6(b). We can see that PLLay consistently improves the accuracies of all baselines. In particular from Table 2 and Figure 6(b), the improvement on CNN is 1.7% \sim 2.8% when the corruption and noise is 0% \sim 5%, and then the improvement goes up to 3.3% when the corruption and noise becomes 10% \sim 15%, and then starts to decrease as the corruption and noise further increases. As discussed in Section 5, this is because although the DTM filtration can robustly capture homological signals up to a moderate amount of corruption and noise, as seen in Figure 2, when the corruption and noise become too much, the topological structure starts to dissolve in the DTM filtration. Also, the accuracies for CNN+P are consistently higher than the accuracies for CNN+P(i), meaning that adding PLLay in the middle of the network indeed further improves the accuracy.



(a) Test accuracy in MNIST data for MLP, MLP+S, MLP+P.



(b) Test accuracy in MNIST data for CNN, CNN+S, CNN+P, CNN+P(i).

Figure 6: Test accuracy in MNIST experiments. PLLay contributes to consistent improvement in accuracy and robustness against noise and corruption. In particular, the improvement on CNN increases up to the moderate level of corruption and noise ($\sim 15\%$), and then start to decrease.

	Noise probability							
	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35
MLP	0.2000 (0.0014)	0.2001 (0.0031)	0.1997 (0.0020)	0.1994 (0.0029)	0.1998 (0.0009)	0.2003 (0.0010)	0.2004 (0.0016)	0.1999 (0.0011)
MLP+S	0.2054 (0.0126)	0.2028 (0.0129)	0.2171 (0.0364)	0.2171 (0.0364)	0.2121 (0.0236)	0.2159 (0.0301)	0.2115 (0.0193)	0.2057 (0.0180)
MLP+P	0.8082 (0.0103)	0.7906 (0.0082)	0.7660 (0.0115)	0.7456 (0.0104)	0.7181 (0.0100)	0.6942 (0.0130)	0.6545 (0.0110)	0.6218 (0.0102)
CNN	0.9466 (0.0116)	0.9247 (0.0152)	0.9053 (0.0195)	0.8791 (0.0255)	0.8224 (0.1474)	0.8323 (0.0298)	0.7963 (0.0331)	0.7401 (0.1293)
CNN+S	0.9412 (0.0182)	0.8881 (0.1612)	0.8142 (0.1900)	0.8142 (0.1900)	0.8197 (0.1473)	0.7777 (0.1875)	0.6580 (0.2622)	0.7195 (0.1778)
CNN+P	0.9511 (0.0140)	0.9249 (0.0308)	0.9095 (0.0329)	0.8941 (0.0305)	0.8619 (0.0366)	0.8480 (0.0173)	0.8087 (0.0396)	0.7668 (0.0319)
CNN+P(i)	0.9449 (0.0343)	0.9319 (0.0290)	0.8965 (0.0471)	0.8873 (0.0143)	0.8577 (0.0349)	0.8285 (0.0515)	0.7954 (0.0516)	0.7543 (0.0553)

Table 3: Test accuracy in ORBIT5K experiments. In each cell, the top number corresponds to the average accuracy of the model at the noise probability, and the bottom number corresponds to the 1 standard deviation of the accuracies. At each column, the model with the best accuracy is bolded.

G.2 Orbit recognition.

For orbit recognition, we use ORBIT5K dataset [Adams et al., 2017, Carrière et al., 2020], a synthetic dataset used as a benchmark in Topological Data Analysis. It consists of a point cloud generated by the following discrete dynamical system: given an initial point $(x_1, y_1) \in [0, 1]^2$ and a parameter $r > 0$, we generate a point cloud $\{(x_n, y_n) \in [0, 1]^2 : n = 1, \dots, N\}$ as

$$\begin{cases} x_{n+1} = x_n + ry_n(1 - y_n) & \text{mod } 1, \\ y_{n+1} = y_n + rx_{n+1}(1 - x_{n+1}) & \text{mod } 1. \end{cases}$$

For comparison with Adams et al. [2017], Carrière et al. [2020], we use parameters $r = 2.5, 3.5, 4.0, 4.1, 4.3$, with random initialization of (x_1, y_1) and $N = 1000$ points in each simulated orbit. We generated 1000 orbits per each value of r , and randomly split the 5000 observations in 70% – 30% training-test sets as in Carrière et al. [2020]. Cross-entropy loss was used to train the network for 100 epochs, using Adam optimizer with mini-batches of size 16. For the noiseless case, the experiment for PointNet is repeated 5 times, and the experiment result for PersLay is from Carrière et al. [2020].

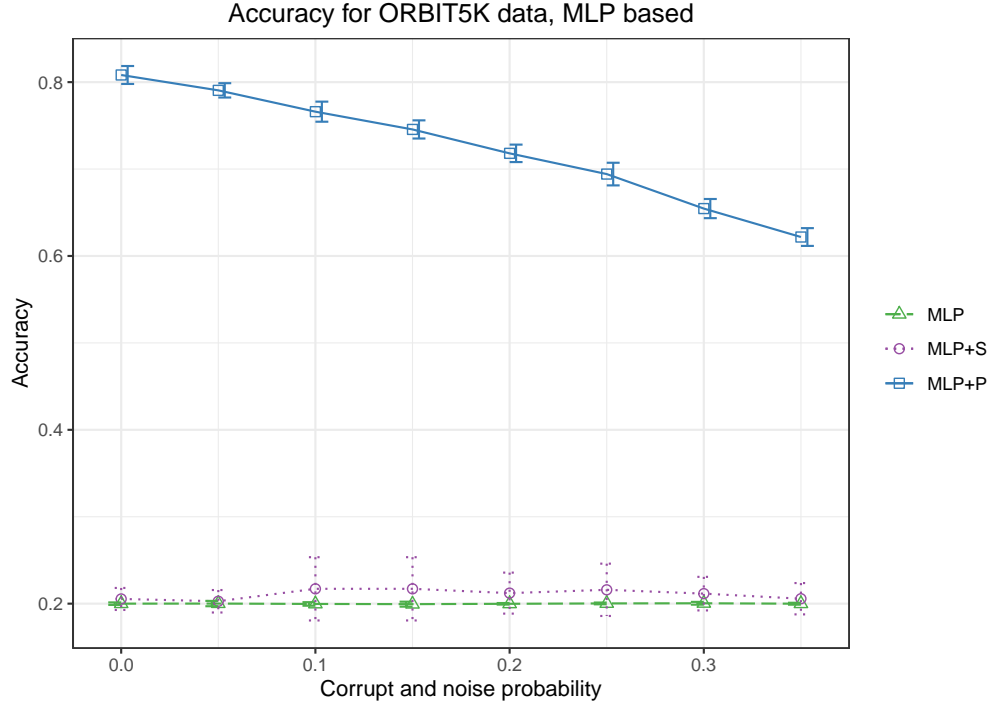
Topological layer. For MLP+P and CNN+P(i), we use one PLLay at the beginning of MLP and CNN models with 64 nodes and affine transformation, which is solely used as the input to MLP or concatenated to the raw input to CNN. We used the empirical DTM filtration in (4), where we define fixed 40×40 points on grid on $[0.0125, 0.9875]^2$ and use X as the empirical data points. We used $m_0 = 0.01$, $K_{\max} = 2$, $T_{\min} = 0.03$, $T_{\max} = 0.1$, $m = 17$. For CNN+P, we additionally use one PLLay after the convolutional layer, with $K_{\max} = 2$, $T_{\min} = 0.05$, $T_{\max} = 0.95$, $m = 18$.

Baselines. For the baselines, models were designed to have simple structures for quick comparisons:

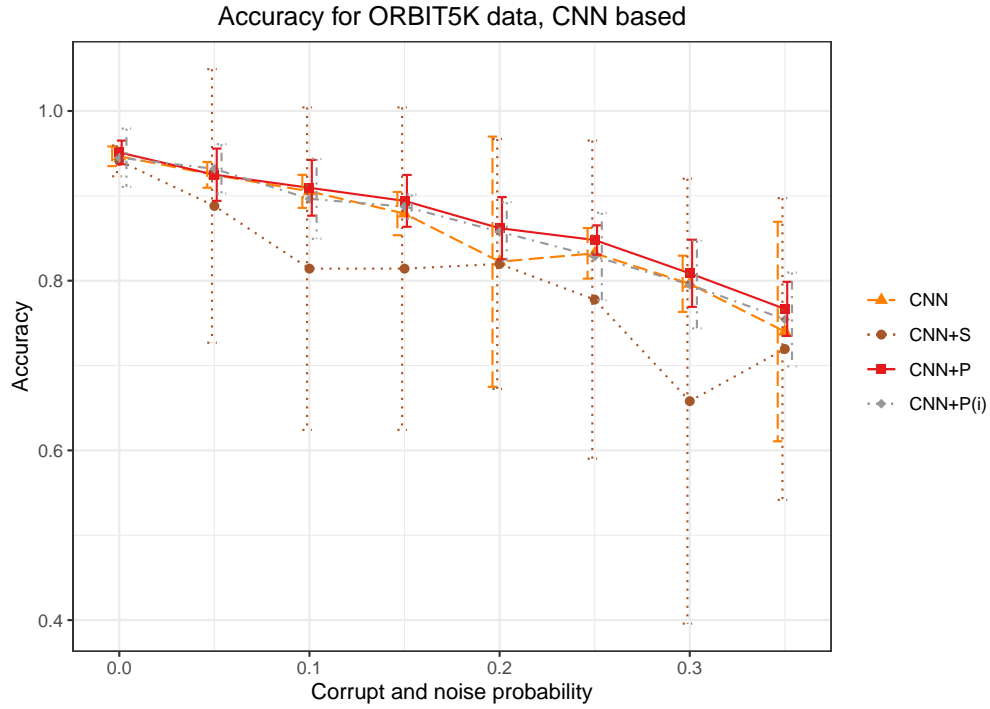
- Vanilla MLP: one hidden layer with 32 units with ReLU activations.
- CNN: two convolution layers followed by two fully connected layers.
- SLay: for comparison with PLLay, one SLay is used with 16 nodes, which is concatenated to the raw input to either MLP or CNN. We used the value $\nu = 0.01$ for the hyperparameter of SLay.

Result. The accuracy results for ORBIT5K data in Figure 4 is represented with 1 standard errors in Table 3 and Figure 7. In Figure 7, the results for MLP, MLP+S, MLP+P are in Figure 7(a), and the results for CNN, CNN+S, CNN+P, CNN+P(i) are in Figure 7(b). From Figure 7(a), we observe that PLLay improves over MLP and MLP+S by a huge margin (42% ~ 60%). In particular, without PLLay, MLP and MLP+S remain at random classifiers, which implies that the topological information is indeed critical for ORBIT5K. In Figure 7(b), PLLay improves over CNN or CNN+S consistently

as well. Moreover, due to the high complexity of ORBIT5K, CNN suffers from high variance at corruption and noise probability 0.2, 0.35, while PLLay can effectively reduce the variance at those simulations and make the models more stable by utilizing robust topological information from the DTM function. Also, the accuracies for CNN+P are almost always higher than the accuracies for CNN+P(i), meaning that adding PLLay in the middle of the network indeed further improves the accuracy.



(a) Test accuracy in ORBIT5K data for MLP, MLP+S, MLP+P.



(b) Test accuracy in ORBIT5K data for CNN, CNN+S, CNN+P, CNN+P(i).

Figure 7: Test accuracy in ORBIT5K experiments. PLLay contributes to consistent improvement in accuracy and robustness against noise and corruption. In particular in (b), when the corruption and noise probability is 0.1, 0.25, 0.35, PLLay effectively reduces the variance of classification accuracy.